Universidade Federal
do Rio de Janeiro
Escola Politécnica

MODELING AND CONTROL DESIGN OF A MULTIFINGERED ROBOT HAND FOR OBJECT GRASPING AND MANIPULATION TASKS

Matheus Ferreira dos Reis

Projeto de Graduação apresentado ao Curso de Engenharia de Controle e Automação da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro em Engenharia de Controle e Automação.

Orientadores: Antonio Candea Leite
Fernando Cesar Lizarralde

Rio de Janeiro
Março de 2016

MODELING AND CONTROL DESIGN OF A MULTIFINGERED ROBOT HAND FOR OBJECT GRASPING AND MANIPULATION TASKS

Matheus Ferreira dos Reis

PROJETO SUBMETIDO AO CORPO DOCENTE DO CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO DA ESCOLA POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO.

Examinado por:

Prof. Fernando Cesar Lizarralde, D.Sc.

Prof. Antonio Candea Leite, D.Sc.

Prof. Ramon Romankevicius Costa, D.Sc.

Prof. Alessandro Rosa Lopes Zachi, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
MARÇO DE 2016

Ferreira dos Reis, Matheus

Modeling and Control Design of a Multifingered Robot Hand for Object Grasping and Manipulation Tasks/Matheus Ferreira dos Reis. – Rio de Janeiro: UFRJ/ Escola Politécnica, 2016.

XII, 105 p.: il.; $29,7$cm.

Orientadores: Antonio Candea Leite
Fernando Cesar Lizarralde

Projeto (graduação) – UFRJ/Escola Politécnica/Curso de Engenharia de Controle e Automação, 2016.

Bibliography: p. 96 – 102.

1. Robot Modeling. 2. Multifingered Robot Hands. 3. Kinematic Control. 4. Visual Servoing Control. I. Leite, Antonio Candea *et al.*. II. Universidade Federal do Rio de Janeiro, Escola Politécnica, Curso de Engenharia de Controle e Automação. III. Título.

À minha querida "pãe" Angela,
pelo apoio, compreensão e carinho
em todos os momentos.

Agradecimentos

Agradeço à minha querida mãe Angela, sem a qual minha trajetória até aqui não teria sido possível, e à minha querida Karina, que traz mais sabor aos meus dias. Aos meus colegas Fernando Coutinho, Maurício Silva, Jonathan Fried, Ivanko Yanque e João Monteiro, e a todos do laboratório, pelo suporte, sugestões, debates e brincadeiras, que contribuíram para tornar agradável nosso ambiente de trabalho. Aos professores Ramon Costa, Liu Hsu, Afonso Celso e a todos os profissionais que contribuíram direta ou indiretamente para minha formação e para a conclusão deste trabalho. Em especial, aos meus orientadores Fernando Lizarralde, pelo suporte, visão e conselhos, e Antonio C. Leite, pela paciência e dedicação em todos os momentos. Agradeço também ao CNPq pelo suporte financeiro.

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Engineer (de Controle e Automação)

MODELING AND CONTROL DESIGN OF A MULTIFINGERED ROBOT HAND FOR OBJECT GRASPING AND MANIPULATION TASKS

Matheus Ferreira dos Reis

March/2016

Advisors: Antonio Candea Leite
Fernando Cesar Lizarralde

Course: Control and Automation Engineering

In this work, we address the problem of kinematic modeling and control design of a multifingered robot hand. Each robot finger is modeled as a parallel manipulator and its kinematic constraints are computed from empirical analysis due to the inherent mechanical complexity of the mechanism. The motion control problem for a grasped object is solved by using the kinematic control approach, which is able to ensure the asymptotic stability of the output tracking error and appropriate prehension of the object. The kinematics-based control scheme is designed to include the contact model in the hand Jacobian matrix, allowing for the simultaneous control of the object position as well as the relative position between the fingers. Additionally, we also deal with the problem of controlling the 3D Cartesian position of an object of known geometry grasped by the robot hand using the visual servoing approach. A video camera and a depth sensor are simultaneously used to provide visual positioning and depth information of the grasped object. The design of the visual servoing controller is also based on the kinematic control approach, and two visual servo controllers are developed in the image space, ensuring the zero-convergence of the output tracking errors and high disturbance-rejection capability. Simulations and experiments are carried out with a three-fingered robot hand and a Microsoft Kinect sensor, illustrating the performance and effectiveness of the proposed methodology.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Robot Hands - Introduction

Most of robot manipulators used for industrial applications are, in general, composed of a large arm equipped with a single gripper or tool for interacting with the environment. These end effectors are custom designed for specific tasks and are able to grasp only a particular class of objects and workpieces. Thus, workpieces with non-parallel faces, spherical objects and fragile plastic parts can be difficult for holding and handling. This drawback can be overcome simply by coupling a tool changer (Fig. 1.1) into the robot arm, allowing for the automatic tool exchanges according to the shape/dimension of the object and, consequently, increasing the robot *versatility* (PRATTICHIZZO e TRINKLE, 2008). Another major drawback usually found in industrial robots is the lack of *dexterity* of their large-scale ma-

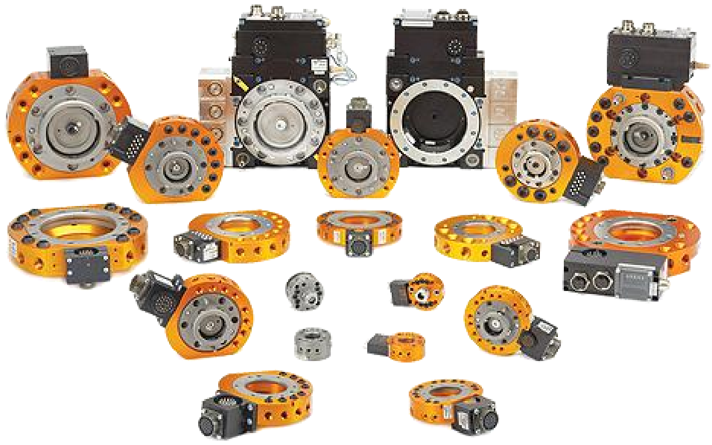


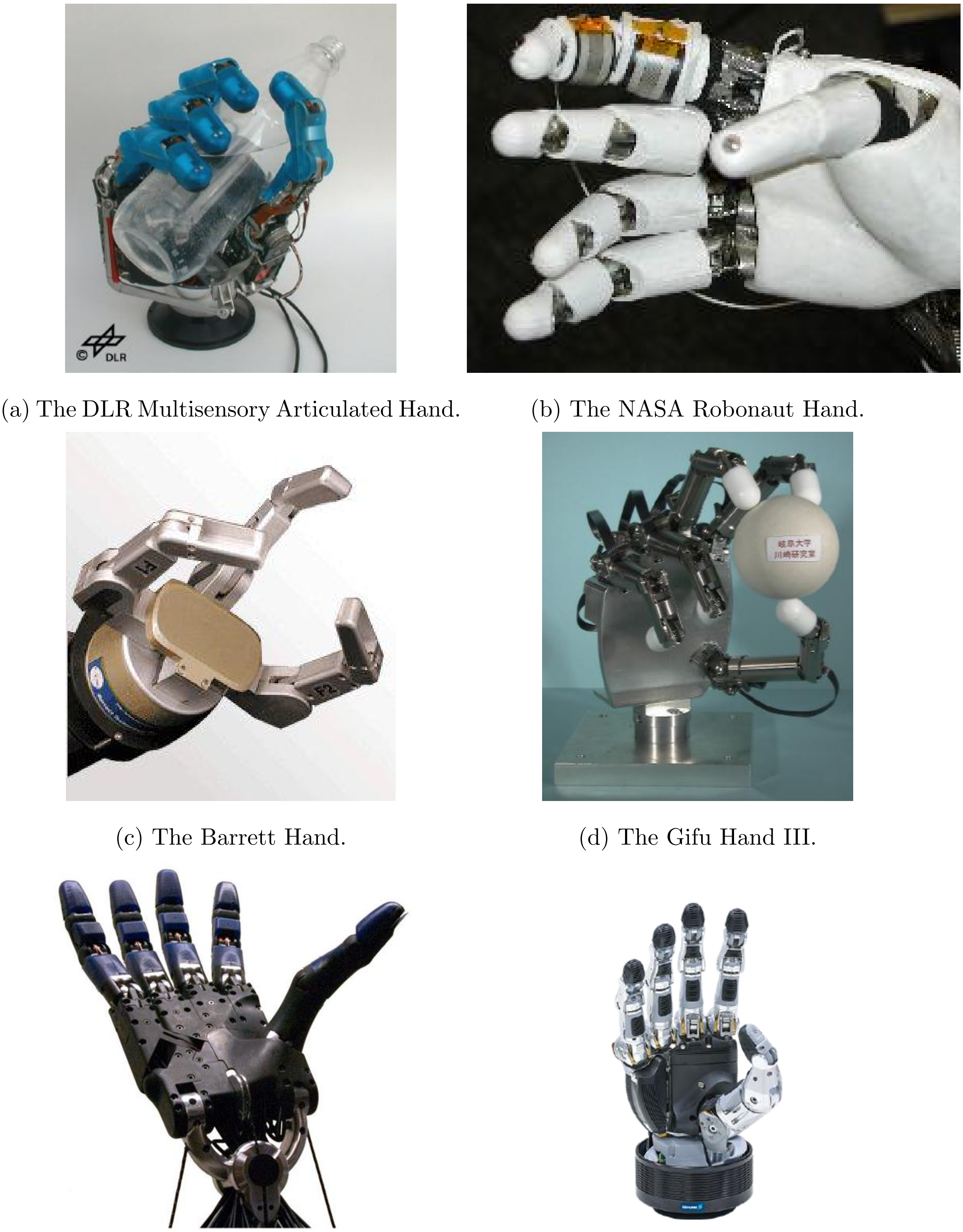
Figure 1.1: Industrial tool changers.

nipulators. This behavior is more evident when the robot arm has to accurately perform precise movements of the payload in assembly tasks. Indeed, fine motions of a grasped object require accurate motions of the robot joints that are difficult to achieve due to the size of the links. On the contrary, a robot hand is able to reconfigure itself for grasping a sort of objects and for achieving a dexterous manipulation in several tasks (BICCHI, 2000). Other capabilities added to robotic systems endowed with robot hands are *perception* of physical properties and *active exploration* of the environment, which are usually unfeasible with simple grippers (ALEOTTI *et al.*, 2014).

The modeling and the design of robot hands is a research topic which has been studied since the seventies, when Okada developed a three-fingered model commanded by a tendon driving system and able to perform nut-opening tasks (OKADA, 1979). In the 80's, two multifingered models were developed by the Jet Propulsion Laboratory (JPL) and the Massachusetts Institute of Technology (MIT) representing a breakthrough in terms of innovation and technology (MELCHIORRI e KANEKO, 2008). Following this trend, a number of robot hands have been developed by universities and research centers around the world for industrial, research and prosthetic applications, such as, the DLR Multisensory Articulated Hand (Fig. 1.2a), the NASA Robonaut Hand (Fig. 1.2b), the Barrett Hand (Fig. 1.2c), the Gifu Hand (Fig. 1.2d) , the Shadow Dexterous Hand (Fig. 1.2e), the Schunk Anthropomorphic Hand (Fig. 1.2f), the i-Limb Ultra Prosthetic Hand (Fig. 1.3a), and the BeBionic Prosthetic Hand (Fig. 1.3b). The main differences among these devices are related to key issues of design, such as number of fingers, kinematic configuration of the fingers around the palm, anthropomorphism and dexterity levels, applicability, type and placement of the actuators and arrangement of the sensors, among others (MELCHIORRI e KANEKO, 2008).

For instance, the Barrett Hand (TOWNSEND, 2000) is composed of a fixed finger and two mobile fingers around the base of the palm, supporting a large variety of grasp configurations. All joints have high-precision position encoders and each finger has torque and tactile sensing.

The Shadow Dexterous Hand (SHADOW ROBOT COMPANY, 2003) is considered one of the most complex anthropomorphic hands ever built. Being equipped



(a) The DLR Multisensory Articulated Hand.

(b) The NASA Robonaut Hand.

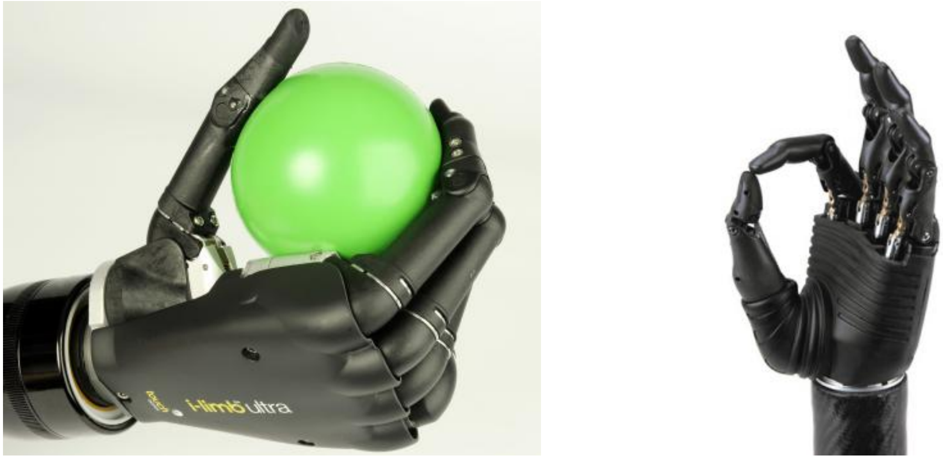
(c) The Barrett Hand.

(d) The Gifu Hand III.

(e) The Shadow Dexterous Hand.

(f) The Schunk Anthropomorphic Hand.

Figure 1.2: Five-fingered robot hands.



(a) The i-Limb Ultra Prosthetic Hand. (b) The BeBionic Prosthetic Hand.

Figure 1.3: Prosthetic robot hands.

with five fingers, the configuration, shape and dimensions of each finger is close to those of the human hand (with 24 DoF, degrees of freedom) and artificial muscles are used for the actuation of the fingers.

The Schunk Anthropomorphic Hand (LIU *et al.*, 2007) consists of three independent aligned fingers and an opposing finger which is analogous to the human thumb, offering good characteristics of dexterity. The finger joints are equipped with magnetic angular sensors and torque sensors.

The i-Limb Hand is considered one of the most capable prosthesis for the hand or upper limb, having five fingers, each with two phalanxes (with 11 DoF). Working as prosthetic device, it is equipped with two electrodes bonded to the patient skin, which collect the myoelectric signals from the muscles and transmit these signals, together with the information from other sensors, to a microprocessor.

Another example of prosthetic hand is the BeBionic Hand, which is similar to the i-Limb Hand in many aspects, such as number of DoF and myoelectric sensors. The main differences are related to the number and types of grasping patterns and cost.

1.2 Modeling and Design of Robot Hands

One of the main aspects related to the complexity and cost of a robot hand is the design of the actuation and transmission system. In terms of the actuator placement, there are two options: (i) *on-site*, where the motors are placed directly inside the joint or hosted inside one of the two links connected by the actuated joint; (ii) *remote*, where the motors are placed outside the links connected by the joint itself (for instance, into the palm or in the forearm). This type of actuation requires a motion transmission system, which could be flexible (e.g., tendons) or rigid (e.g., gear trains) according to the transmission element to be adopted.

An appropriated sensory system is also a key part of the project of a robot hand. It provides all necessary information for the operations to be correctly performed. From a computational point of view, sensors are the primary channels of data needed by the robot operational system to generate a description of the environment around and of the robot itself. This is essential when we want the robot to operate in non-structured environments such as our everyday world, rather than operating in a controlled environment such as in manufacturing applications (HUTCHINSON *et al.*, 1996).

Sensors can be classified in two different types in robotics literature: *proprioceptive* and *exteroceptive* sensors (SICILIANO *et al.*, 2009). The first type is responsible for measuring the internal state of the robot manipulator. As an example of *proprioceptive* sensor, *encoders* are utilized to recover angular position data of each mechanical joint. This information can be utilized together with a kinematic/dynamic model of the manipulator in operational space control schemes. *Exteroceptive* sensors are capable of providing information about the surrounding environment. As examples, we can cite force and torque sensors, necessary for force control strategies that account for the interaction between the end-effector and the task surface (LUIGI VILLANI, 2008).

A multifingered robot hand can be modeled as a set of small robots fixed on a common base or palm. Indeed, the kinematic and dynamic models of a robot hand with on-site actuation are quite similar to those of a typical industrial robot, which simplifies the modeling problem. On the contrary, remote actuation adds some challenges to the control design, such as, the compliance of the transmission system and non-negligible dynamic effects (*e.g.*, friction and backlash). Meanwhile, most of control strategies developed for coordinated manipulation between robot manipulators can be immediately extended and applied to a multifingered robot hand grasping an object (TINÓS *et al.*, FREITAS *et al.*, 2006, 2011).

However, the design of high-level control schemes for a robot hand must take into account the interaction between the fingers and the object. This mutual interaction depends on several aspects, such as modeling of contact points (*e.g.*, compliance, friction), force/torque control at the contact points, the mobility of the fingers and the contacts (*e.g.*, fixed, rolling), as well as planning algorithms for grasping/manipulating of objects (KAO *et al.*, YOSHIKAWA, 2008, 2010). In this context, several control strategies for robot hands operating under a wide range of uncertainties have been developed using adaptive, robust, hybrid and artificial intelligence techniques (TAKAHASHI *et al.*, ENGEBERG e MEEK, YIN *et al.*, ARIMOTO, ZHAO e CHEAH, 2008, 2013, 2003, 2004, 2009).

1.3 Visual Servoing for Control of Robot Hands

Since the early 70's cameras have been used to obtain geometrical information on the surrounding environment without physical interaction, likewise the human sense of vision. Typical applications include surveillance, monitoring and inspection. The concept of using feedback information extracted from a visual sensor to control the motion of a robotic system with respect to the pose of a target object is referred to as *visual servoing* or vision-based robot control. In this context, the visual system can be integrated by a single or multiple cameras, mounted in fixed (eye-to-hand) or

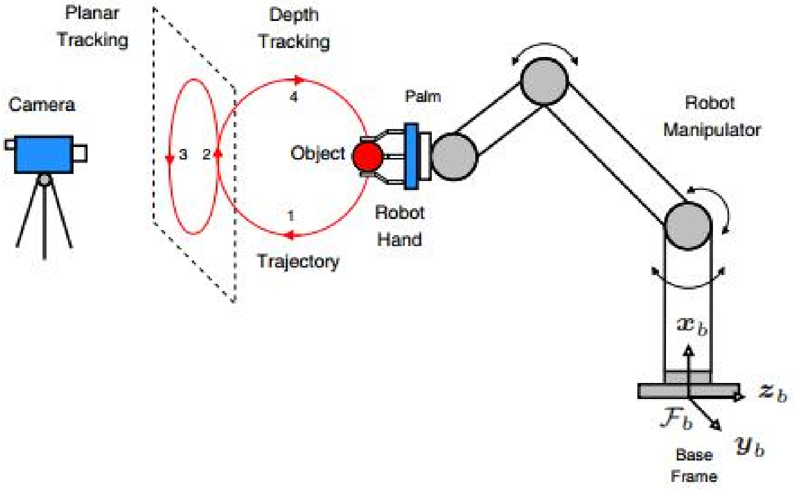


Figure 1.4: Typical visual servoing scheme.

moving (eye-in-hand) configurations. The visual servoing techniques can be widely classified into the following categories: position-based, image-based or hybrid, which combines the common characteristics of the first two. The main differences between the position-based visual servoing (PBVS) and image-based visual servoing (IBVS) are related to the coordinates space where the output error is computed (e.g., Cartesian or image), and how the current pose of the target object is obtained (e.g., pose estimate or feature measurements). The main benefits of using the visual information directly into a feedback control loop are twofold: lower sensitive to camera calibration errors and there is no need to estimate the object pose with respect to the camera frame in real-time (CHAUMETTE e HUTCHINSON, 2008). Vision also offers an interesting solution as a non-contact robot sensor, since it provides a great amount of information about the surrounding world in a way similar to the human vision (HUTCHINSON *et al.*, 1996). The disadvantage of using vision to extract useful information from the surrounding environment is the need for image processing techniques, which are usually computationally demanding and not easy to implement. Despite of this drawback, integrating cameras into a robotic system can not only radically increase the number of feasible tasks but can also increase the task accuracy, if a visual feedback loop is used for correcting the position of the robot manipulator (SHIRAI e INOUE, 1973).

Inspired by the natural way in which humans pick up different objects and manipulate them using integrated vision and touch sensing to control the grasping motion, a number of vision-based object manipulation tasks using robot hands have been proposed (SEITZ, 1999). The idea is to employ the visual servoing framework to control the position of an unknown object grasped by a multifingered robot hand (LIPPIELLO *et al.*, 2013). The vision system could also be used to overcome the lack of knowledge about the shape or geometry of the object. From the combination of vision and force data at control level, integrated manipulation frameworks have been developed to carry out common daily tasks such as opening doors, pushing buttons or turning knobs (PRATS *et al.*, 2008). Tactile sensing have been combined to 3D stereo vision or high-speed vision systems to improve the performance of the object manipulation tasks under dynamic changes in the working environment (HONDA *et al.*, NAMIKI *et al.*, 1998, 1999). Robust visual servoing methods

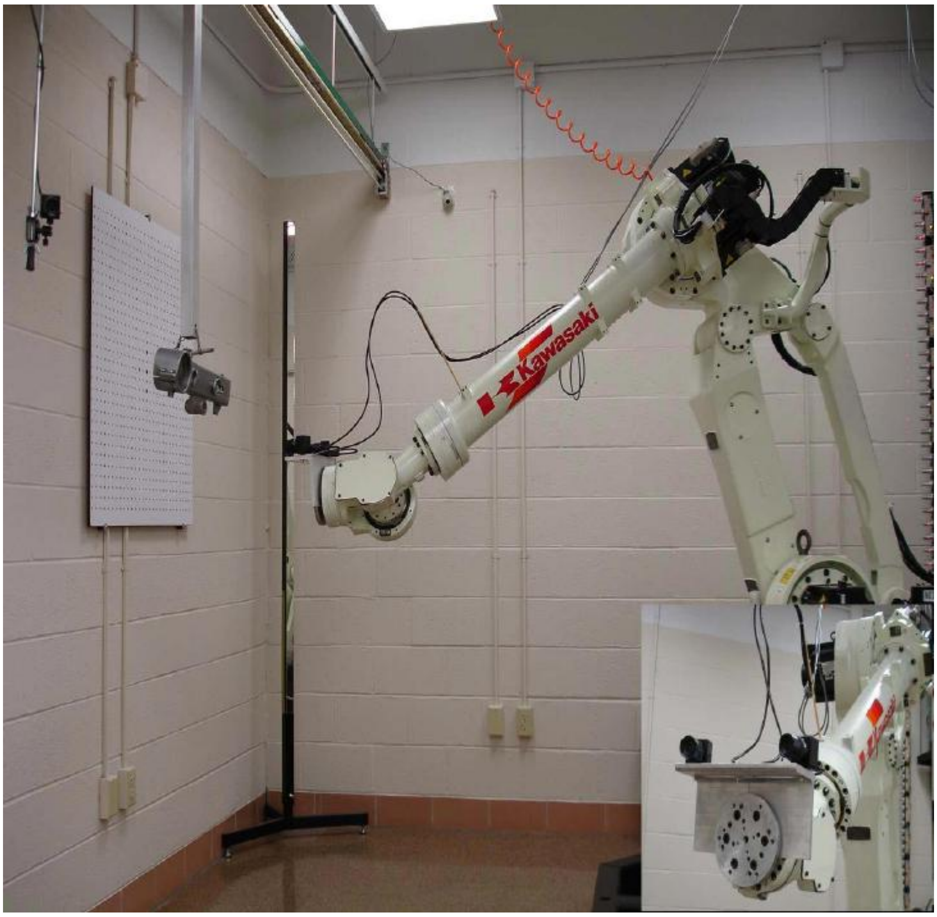


Figure 1.5: Eye-in-hand visual servoing scheme with a Kawasaki robot manipulator. From ViGIR Lab, University of Missouri.

for object manipulation have also been proposed to compensate for the effects of frictional forces at the fingertips or to deal with the temporary loss of information due to occlusion or object disappearance (YOKOKOHJI *et al.*, KAWAMURA *et al.*, 1999, 2013). Some works have considered the dynamic model of the robot hand and the grasping force of the fingertips to implement control schemes based on neural networks approach and optimal control theory (ZHAO e CHEAH, JARA *et al.*, 2009, 2014).

1.4 Overview of this work

In this work, we present a kinematic framework for modeling and control design of a three-fingered robot hand. We consider two different types of actuation, both usually found in many types of robot hand designs: (i) on-site actuation and (ii) remote actuation. In many cases of remote actuation, each robot finger can be considered as a parallel manipulator subjected to kinematic constraints. A simplified approach is proposed where these constraints can be computed from an empirical model for the movement of the fingers. This approach is motivated by the inherent mechanical complexity of many transmission mechanisms usually found in robot hands.

The kinematic control approach is used to tackle the position control problem for the fingers in both cases, ensuring the asymptotic stability of the output tracking error. The problem of controlling the grasping motions for a robot hand during the manipulation task of an object is conceptually analogous to the control problem of multiple manipulators cooperating to perform some coordinated task, for example, lifting a box. Thus, well-known modeling and control strategies usually applied to cooperative robot manipulators can also be naturally extended to multifingered robot hands (WEN e WILFINGER, FREITAS *et al.*, 1999, 2011). To solve the problem of grasping and manipulating an object, contact and relative position models for the fingers are developed, and a kinematics-based control scheme is designed to allow for a successful motion control and prehension of the grasped object. The key idea behind this solution is based on the previous work on modeling and control design for general robotic systems subject to kinematic constraints (WEN e WILFINGER, 1999).

We also address the problem of controlling the 3D Cartesian position of a multifingered robot hand for object grasping tasks using a visual feedback loop. The solution presented in this work covers a vision system composed of calibrated video cameras and depth sensors. The control design is also based on the kinematic control approach and two visual servoing schemes are derived to ensure the asymptotic stability and the zero-convergence of the output tracking errors: an image-based visual servoing method (IBVS) and a hybrid visual servoing method (HVS), which combines image and Cartesian coordinates for control.

Experimental tests, performed with the 3-Finger Adaptive Robot Gripper from

Robotiq (Fig. 2.4a), illustrate the applicability of the proposed methodology for object grasping and manipulation tasks by general multifingered robot hands. The

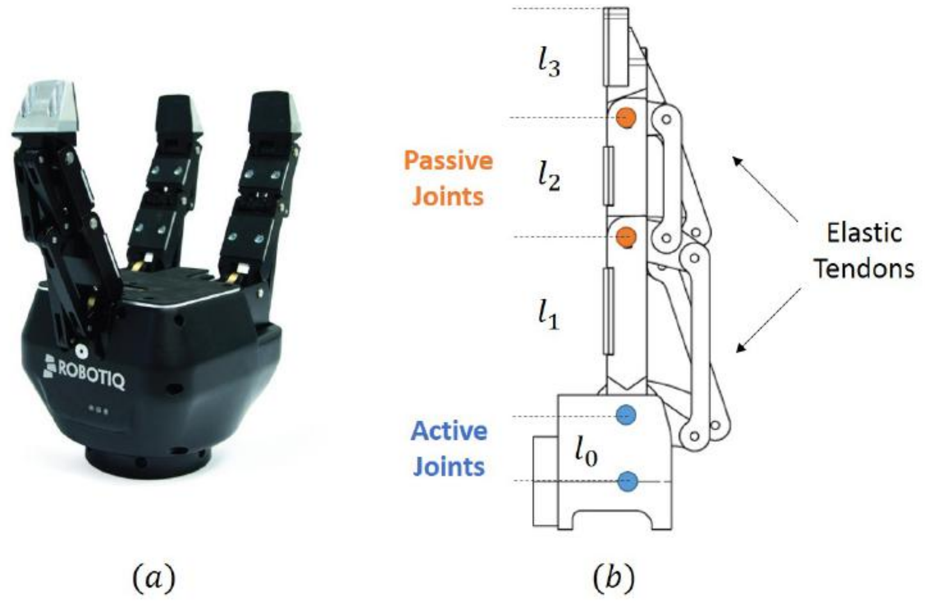


Figure 1.6: Multifingered robot hand: (a) 3-Finger Adaptive Robot Gripper; (b) Articulated finger with elastic tendons. Courtesy from Robotiq.

video camera is also used to validate the empirical model previously developed for the movement of the fingers of the 3-Finger Adaptive Robot Gripper. This is done by observing how the finger phalanges move when the robot hand actuators are controlled, and then constructing functions that model these relationships. Computer simulations illustrate more general applications of the presented theoretical framework, for example in the case of on-site actuation, where generally the robot hand has more DoF than in the case of remote actuation.

Chapter 2

Kinematic Modeling of a Robot Hand

In this chapter, we develop the kinematic models used in this work. In particular, we develop kinematic models for: a general three-fingered robot hand, the finger kinematic constraints introduced by the presence of *remote actuation* (when the joint is driven by actuators placed outside the links connected by the joint [5]) and the grasp constraints induced by the contact between finger and object. The first section introduces the terminology and notation used in the work. In the next sections, two different approaches are considered: in section 2.2, we develop a kinematic model for a three-fingered robot hand with on-site actuation.

In section 2.3, an introduction to the theory of parallel mechanisms is developed. Motivated by models of grippers commonly found in industry, two different modeling approaches are proposed for the case where the fingers are closed-chain, under-actuated mechanisms driven by remote actuation. In the first case, the kinematic constraints on the joint velocities are derived analytically. This approach is specially suitable for robot hands with simple mechanical structures / transmission systems. In the second case, the kinematic constraints are derived by using some technique for extracting information about the relations between the finger joints. Calibrated video cameras could be used with colored targets placed in specific points in the structure of the fingers, so that the movement relations could be empirically derived.

Section 2.4 provides a brief introduction to grasping kinematics, and a general model for the kinematic constraints that characterize the grasping of an object is derived.

Section 2.5 provides an introduction to the relative kinematics of the robot hand, important for the future development of a suitable prehension scheme for the grasped object. This scheme ensures the validity of the methods developed in the previous section.

Finally, in section 2.6, we employ the methodology developed in the previous sections to find a proper model for the 3-Finger Adaptive Robot Hand, from Robotiq.

2.1 Terminology and Notation

In this first section, we introduce the terminology and notation used in this work. Fig 2.1 illustrates the adopted conventions for the finger angles, link lengths and coordinate frames used in Chapter 2.

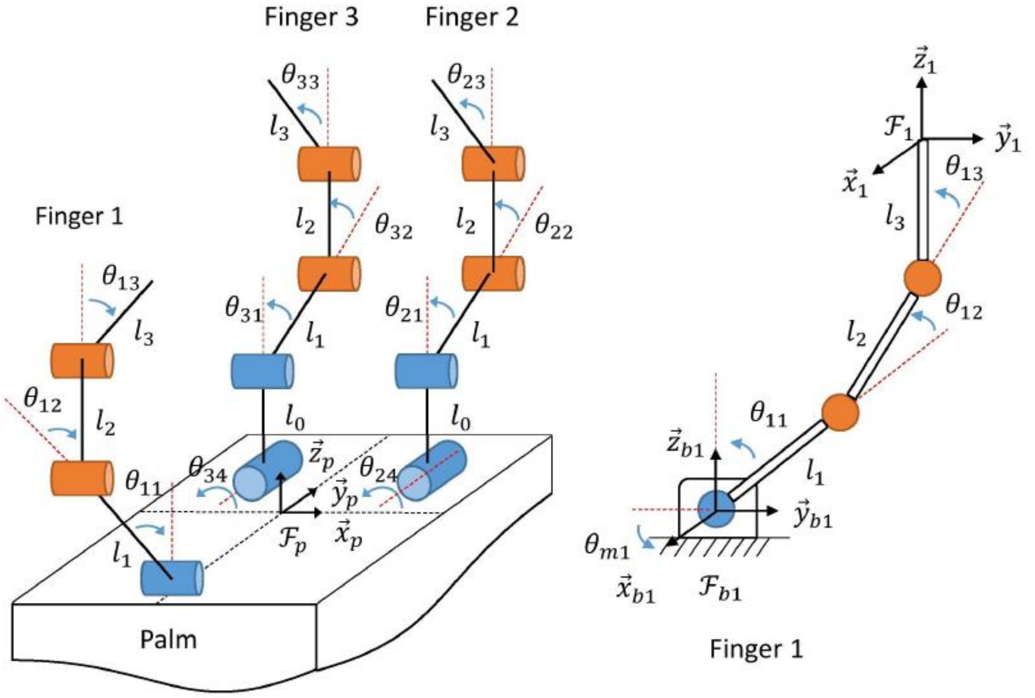


Figure 2.1: Convention for the links and joints motions of the robot hand.

θ_{kj} : angle of the j-th joint for the k-th finger;

θ_{ka} : angle of the k-th finger driver motor;

θ_{4a} : angle of the motor that drives the scissor motion;

$\mathcal{F}_p$: palm coordinate system defined by the unit vectors of the frames axes $\vec{x}_p$, $\vec{y}_p$ and $\vec{z}_p$. This is the reference frame in which all vectors will be referred to, unless otherwise stated;

$\mathcal{F}_{bk}$: coordinate system defined in the base of k-th finger;

$\mathcal{F}_k$: fingertip coordinate system defined by the unit vectors of the frames axes $\vec{x}_k$, $\vec{y}_k$ and $\vec{z}_k$;

$\mathcal{F}_o$: object coordinate system defined by the unit vectors of the frames axes $\vec{x}_o$, $\vec{y}_o$ and $\vec{z}_o$;

l_i : length of the i-th phalange where $i = 1, 2, 3$ represent the proximal, medial and distal phalanges respectively;

l_0 : distance between the pitch axis and the axis of the first joint, for fingers 2 and 3;

R_a : rotation matrix of an arbitrary frame $\bar{E}_a$ with respect to the palm frame $\bar{E}_p$;

p_i : position vector of the i-th fingertip frame with respect to the palm frame defined in terms of its x, y, z coordinates as $p_i = (p_{ix},\, p_{iy},\, p_{iz})$.

2.2 Forward and Differential Kinematics of a Multi-fingered Robot Hand

In this section, we develop the kinematic modeling for a three-fingered robot hand. We follow an intuitive and simplified approach, where the robot can be considered as a multi-robot system composed of a set of robot fingers fixed on a common base or palm. Since each finger can be modeled as a single and independent mechanism, it can be controlled separately. The palm coordinate system denoted by $\mathcal{F}_p$ is the inertial frame in which all vectors and matrices will be referred to hereafter. Consider that the robot hand is composed of n articulated fingers coupled to the palm, each one with n_k revolute joints $(k = 1, 2, ..., n)$. Let the pair $\{p_k\,, R_k\}$ be the position vector and the rotation matrix, which represent respectively the location and the orientation of the k-th fingertip frame $\mathcal{F}_k$ with respect to the palm frame $\mathcal{F}_p$. The pose of the k-th finger can be obtained from the *forward kinematics* map as:

$$p_k = p_k(\theta_k)\,, \qquad R_k = R_k(\theta_k)\,, \tag{2.1}$$

where $\theta_k \in \mathbb{R}^{n_k}$ is the joint position vector for the k-th finger. The *forward kinematics* map can also be expressed in terms of the vector

$$\mathbf{x}_k = \begin{bmatrix} p_k \\ \phi_k \end{bmatrix} = h_k(\theta_k)\,, \tag{2.2}$$

where ϕ_k denotes a particular choice of variables for the representation of the finger orientation, such as a particular convention of Euler angles.

The coordinate transformation between the frames $\mathcal{F}_k$ and $\mathcal{F}_p$ for the k-th finger can be written in terms of the *homogeneous transformation* matrix as (SICILIANO *et al.*, 2009):

$$T_k(\theta_k) = \begin{bmatrix} R_k(\theta_k) & p_k(\theta_k) \\ 0^{\mathsf{T}} & 1 \end{bmatrix}. \tag{2.3}$$

Let $v_k \in \mathbb{R}^6$ be the velocity vector of the k-th fingertip frame $\mathcal{F}_k$ with respect to the palm frame $\mathcal{F}_p$. It can be related to the joint velocity vector $\dot{\theta}_k \in \mathbb{R}^{n_k}$ by the *differential kinematics* equation as:

$$v_k = \begin{bmatrix} \dot{p}_k \\ \omega_k \end{bmatrix} = \underbrace{\begin{bmatrix} J^k_{f_P} \\ J^k_{f_O} \end{bmatrix}}_{J^k_f} \dot{\theta}_k\,. \tag{2.4}$$

where the pair $\{\dot{p}_k\,, \omega_k\}$ denotes the linear and angular velocity components of v_k and $J^k_f \in \mathbb{R}^{6\times n_k}$ is the geometric Jacobian of the k-th finger, composed of linear and angular parts ($J^k_{f_P} \in \mathbb{R}^{3\times n_k}$ and $J^k_{f_O} \in \mathbb{R}^{3\times n_k}$, respectively).

The angular velocity vector ω_k can be related to the derivative of the Euler angles $\dot{\phi}_k$, by means of

$$\dot{\phi}_k = T_e(\phi_k)\,\omega_k\,, \tag{2.5}$$

where T_e is a transformation matrix, whose expression is dependent on the particular choice of representation for finger orientation (SICILIANO *et al.*, 2009).

Analogously to (2.4), we can write the differential kinematics equation for each finger is terms of the direct derivative of the forward kinematics map (2.2). Using

(2.5), the relation between $\dot{h}_k$ and v_k is given by:

$$\dot{h}_k = T_A(\phi_k)\, v_k\,, \qquad T_A(\phi_k) = \begin{bmatrix} I_3 & 0 \\ 0 & T_e(\phi_k) \end{bmatrix}. \tag{2.6}$$

By direct differentiation of the forward kinematic map (2.2), we get:

$$\dot{\mathbf{x}}_k = \frac{\partial h_k(\theta_k)}{\partial \theta_k}\, \dot{\theta}_k = J_A^k(\theta_k)\, \dot{\theta}_k\,, \tag{2.7}$$

where $J_A^k \in \mathbb{R}^{6\times n_k}$ is the so-called *analytical Jacobian matrix* of the k-th finger.

Combining (2.6) with (2.4) and comparing with (2.7), it is easy to see that the relation between the analytical and the geometric finger Jacobian matrices is given by:

$$J_A^k = T_A(\phi_k)\; J_f^k\,. \tag{2.8}$$

2.3 Finger Kinematic Constraints

In this section, we consider the common case where the motions of the robot fingers are subject to kinematic constraints on velocity. As mentioned earlier, many robot hands are remote actuated, meaning that the fingers are indirectly driven by motors by means of a transmission system. The transmission system can be composed of springs, gears, complex parallel mechanisms, or even combinations of these. In some robot hands, the fingers can be considered as closed-chain mechanisms composed of active and passive joints. Indeed, passive joints are indirectly controlled by the motion of actuated (or active) joints. In general, if $\theta_k \in \mathbb{R}^{n_k}$ is the vector of the n_k joint angles of the k-th finger and there are m_k constraint kinematic equations (in the k-th finger) relating these angles, we can express these constraints as:

$$g_{kj}(\theta_k) = 0\,, \quad j = 1, 2, ..., m_k\,. \tag{2.9}$$

Suppose that θ_k can be partitioned into: $\theta_k = \begin{bmatrix} \theta_{ka}^{\mathsf{T}} & \theta_{kp}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$. where θ_{ka} and θ_{kp} are vectors containing the angles of the active and the passive joints of the k-finger, respectively. In general, the number of passive joints is the same as the number of constraint equations, such that $\theta_{kp} \in \mathbb{R}^{m_k}$ and $\theta_{ka} \in \mathbb{R}^{n_k - m_k}$.

Consider that it is possible to obtain an analytical expression for the passive joints θ_{kp}:

$$\theta_{kp} = c_k(\theta_{ka})\,, \tag{2.10}$$

where $c_k(.)$ is a map of the active joint angles of the k-th finger to its passive joint angles. The relation $c_k(.)$ can be found either analytically or empirically, if it is possible to observe how the passive joints move with respect the motion of the active joints in finger k. For example, if all the joints (including the passive ones) are equipped with encoders, than the problem of finding $c_k(.)$ is trivial, since one can easily infer the relations by observing the encoder data. Another possibility is to use the encoder data and curve fitting algorithms with a calibrated video camera and image processing techniques to find out the mentioned relations.

In the assumption of known and differentiable $c_k(.)$ functions, if we differentiate 2.10 directly, we obtain:

$$\dot{\theta}_{kp} = \frac{\partial c_k(\theta_{ka})}{\partial \theta_{ka}} \dot{\theta}_{ka} = J_c^k(\theta_{ka})\,\dot{\theta}_{ka}\,. \tag{2.11}$$

Now, suppose that the fingertip velocities v_k in 2.4 are dependent on both the active and the passive joint velocities of the mechanism. Thus, we split the differential kinematics equation in terms of the active and passive joints:

$$v_k = \begin{bmatrix} J_{fa}^k & J_{fp}^k \end{bmatrix} \begin{bmatrix} \dot{\theta}_{ka} \\ \dot{\theta}_{kp} \end{bmatrix} = J_{fa}^k\,\dot{\theta}_{ka} + J_{fp}^k\,\dot{\theta}_{kp} \tag{2.12}$$

Now, we can use 2.11 to derive the differential kinematics equation relating the fingertips velocity to the active joints velocity:

$$v_k = (J_{fa}^k + J_{fp}^k\,J_c^k)\,\dot{\theta}_{ka}\,. \tag{2.13}$$

We can also write 2.13 in the following way, which will be helpful in the next sections:

$$v_k = \underbrace{\begin{bmatrix} J_{fa}^k & J_{fp}^k \end{bmatrix}}_{J_f^k} \underbrace{\begin{bmatrix} \mathsf{I} \\ J_c^k \end{bmatrix}}_{\tilde{J}_c^k} \dot{\theta}_{ka} \tag{2.14}$$

Thus, one can always express the Jacobian matrix in the differential kinematics equation of a constrained mechanism as a product of two matrices:

$$v_k = J_f^k\,\tilde{J}_c^k\,\dot{\theta}_{ka} \tag{2.15}$$

where J_f^k is the well-known geometric Jacobian of the k-finger and $\tilde{J}_c^k$ is an augmented constraint Jacobian of the mechanism.

Proposition 1 *In the particular case where the fingertips pose does not depend directly on the active joints angles θ_{ka}, but only on the passive joints angles θ_{kp} ($\theta_k = \theta_{kp}$):*

$$\text{(i) } J_f^k = J_{fp}^k \ ,$$
$$\text{(ii) } \tilde{J}_c^k = J_c^k$$

and thus the differential kinematics equation 2.15 reduces to

$$v_k = J_f^k \, J_c^k \, \dot{\theta}_{ka} \, . \tag{2.16}$$

Proof 1 *See Appendix A - A.1.*

Back to equation (2.11), we can also express the constraint Jacobian matrices in another way. First, defining the vector of active joints of the hand as $\theta_a = [\, \theta_{1a} \ \theta_{2a} \ \ldots \ \theta_{na} \,]^\mathsf{T}$, we can combine the constraint Jacobian matrices with appropriate null matrices to express the active to passive joint velocity transformation for the k-th finger with respect to θ_a:

$$\dot{\theta}_{kp} = \underbrace{\begin{bmatrix} 0 & \ldots & J_c^k & \ldots & 0 \end{bmatrix}}_{J_C^k} \underbrace{\begin{bmatrix} \dot{\theta}_{1a} \\ \dot{\theta}_{2a} \\ \vdots \\ \dot{\theta}_{na} \end{bmatrix}}_{\dot{\theta}_a} = J_C^k \, \dot{\theta}_a \, , \tag{2.17}$$

where $J_C^k \in \mathbb{R}^{n_k \times n_m}$ are the augmented constraint Jacobian matrices.

If Proposition 1 is the case, then we can rewrite equation 2.15 as:

$$v_k = \underbrace{J_f^k \, J_C^k}_{J_F^k} \, \dot{\theta}_a \, , \tag{2.18}$$

which will be useful in the beginning of next section. The matrix $J_F^k \in \mathbb{R}^{6 \times n_m}$ will be referred as the augmented finger Jacobian.

In Appendix B, we cover the case of non-integrable constraint kinematic equations 2.9.

2.4 Multifinger Grasp Kinematics

In this section, we present the kinematic modeling for a multi-fingered robot hand grasping and/or manipulating a given object. Our goal is to find the differential kinematics equation relating the velocity of a reference frame fixed in the manipulated object and the velocities of the active joints of the mechanism. Here, we suppose that the object is held firmly between the palm and the fingertips, while it follows a desired reference trajectory. According to (2.15), we can stack the linear and angular velocity vectors v_k for all n fingers:

$$\underbrace{\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}}_{v_h} = \underbrace{\begin{bmatrix} J_f^1 & 0 & \dots & 0 \\ 0 & J_f^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & J_f^n \end{bmatrix}}_{J_f} \underbrace{\begin{bmatrix} J_C^1 \\ J_C^2 \\ \vdots \\ J_C^n \end{bmatrix}}_{J_c} \dot{\theta}_a \,. \tag{2.19}$$

In shorthand notation:

$$v_h = \underbrace{J_f\, J_c}_{J_h}\, \dot{\theta}_a = J_h\, \dot{\theta}_a \,. \tag{2.20}$$

where $v_h \in \mathbb{R}^{6n}$ is the velocity vector of the robot hand and $J_h \in \mathbb{R}^{6n \times n_m}$ is the well-known *hand Jacobian* matrix, written in terms of the product of matrices $J_f \in \mathbb{R}^{6n \times N}$ and $J_c \in \mathbb{R}^{N \times n_m}$, where $N = \sum_{k=1}^{n} n_k$ and n_m is the total number of active joints of the hand. The hand Jacobian matrix J_h can also be written in terms of the augmented finger Jacobian matrices J_F^k, introduced in the end of last section:

$$J_h = \begin{bmatrix} J_F^{1\,\mathsf{T}} & J_F^{2\,\mathsf{T}} & \dots & J_F^{n\,\mathsf{T}} \end{bmatrix}^{\mathsf{T}} . \tag{2.21}$$

Now, let r_{ko} be the position vector of the k-th contact point with respect to the object frame $\mathcal{F}_o$, where it is constant regardless of the object's motion. Let also R_{po} be the rotation matrix of the object frame with respect to the palm frame $\mathcal{F}_p$. According to (WEN e WILFINGER, CACCAVALE e UCHIYAMA, 1999, 2008), the object velocity vector v_o can be related to the object velocity vector at the k-th contact point v_{ck} by:

$$v_{ck} = A_k(r_{ko})\, v_o \,, \qquad A_k(r_{ko}) = \begin{bmatrix} I_3 & -S(R_{po}\, r_{ko}) \\ 0 & I_3 \end{bmatrix} , \tag{2.22}$$

where $S(.) \in \mathbb{R}^{3\times 3}$ is the skew-symmetric matrix operator. The matrix $A_k \in \mathbb{R}^{6\times 6}$ is the Adjoint matrix related to the k-th contact point. Now, under the assumption of firm contact (the k-th finger never loses contact with the object, for all $k = 1, 2, \dots, n$), the velocity vector v_{ck} can be parameterized in terms of a generic velocity vector w_k using:

$$v_k + H_k\, w_k = v_{ck}\,, \tag{2.23}$$

where the columns of the constraint matrix $H_k \in \mathbb{R}^{6\times n_r}$ represents the n_r degrees of freedom for the k-th fingertip at the contact point.

There are some interpretations for the velocity vector w_k. Under the assumption of a firm contact, the whole system composed of fingers and object can be thought of a parallel mechanism, where the vector w_k represents the angular velocity of the virtual joint formed between the k-th finger and the grasped object.

Thus, the constraint matrix H_k accept or reject velocity components of w_k at the contact point, transmitting only the desired components of the motion. Some examples of types of contacts and corresponding values of the constraint matrix H_k can be found in (WEN e WILFINGER, 1999).

Now, we can stack (2.22) and (2.23) for all fingers to obtain:

$$\underbrace{\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}}_{v_h} + \underbrace{\begin{bmatrix} H_1 & 0 & \dots & 0 \\ \vdots & H_2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & H_n \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}}_{w} = \underbrace{\begin{bmatrix} v_{c1} \\ v_{c2} \\ \vdots \\ v_{cn} \end{bmatrix}}_{v_c},$$

and

$$\underbrace{\begin{bmatrix} v_{c1} \\ v_{c2} \\ \vdots \\ v_{cn} \end{bmatrix}}_{v_c} = \underbrace{\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix}}_{A} v_o\,.$$

Writing the same equations in a shorthand notation, according to the definitions above:

$$v_h + H\, w = v_c \tag{2.24}$$

$$v_c = A\, v_o\,, \tag{2.25}$$

where:

$$w = [\, w_1^{\mathsf{T}} \;\; w_2^{\mathsf{T}} \;\; \ldots w_n^{\mathsf{T}} \,]^{\mathsf{T}} \,, \qquad H = diag(H_1 \,, \ldots \,, H_n) \,,$$
$$A = [\, A_1^{\mathsf{T}} \;\; A_2^{\mathsf{T}} \;\; \ldots \;\; A_n^{\mathsf{T}} \,]^{\mathsf{T}} \,, \qquad v_c = [\, v_{c1}^{\mathsf{T}} \;\; v_{c2}^{\mathsf{T}} \;\; \ldots \;\; v_{cn}^{\mathsf{T}} \,]^{\mathsf{T}} \,.$$

Then, combining (2.20) with (2.24) and (2.25), we obtain the following differential kinematics equation:

$$J_h \, \dot{\theta}_a + H \, w = A \, v_o \,. \tag{2.26}$$

Pre-multiplying both sides of (2.26) by the Moore-Penrose *left* pseudo-inverse matrix A^+ and the annihilating matrix $\tilde{A}$, such that $A^+ A = I$ and $\tilde{A} \, A = 0$, we obtain the *multifinger kinematic model* as:

$$v_o = (A^+ J_h) \, \dot{\theta}_a + (A^+ \, H) \, w \,, \tag{2.27}$$

$$0 = (\tilde{A} \, J_h) \, \dot{\theta}_a + (\tilde{A} \, H) \, w \,, \tag{2.28}$$

where A is of full column rank. Notice that, from (2.27) and (2.28), the variable w can be interpreted as the velocity vector of the contact point. Indeed, during the grasping situations, the contact point between the fingertip and the object can be modeled as a virtual rotational/prismatic joint performing rolling/sliding motion (WEN e WILFINGER, 1999). If we specify $\dot{\theta}_m$, it is possible to solve (2.28) for w, provided that the matrix $(\tilde{A} H^{\mathsf{T}})$ is non-singular. Thus, we can rewrite (2.27) as:

$$v_o = \underbrace{(G_o \, J_h)}_{J_o} \, \dot{\theta}_a \,, \qquad G_o = A^+ [\, I - H \, (\tilde{A} \, H)^+ \tilde{A} \,] \,, \tag{2.29}$$

where $J_o \in \mathbb{R}^{6 \times n_m}$ is the object Jacobian matrix, written as the product of a *grasp matrix* G_o and the hand Jacobian matrix J_h.

Equation (2.29) can also be written in terms of the position and orientation parts of the object Jacobian matrix, $J_{Po} \in \mathbb{R}^{3 \times n_m}$ and $J_{Oo} \in \mathbb{R}^{3 \times n_m}$, respectively:

$$v_o = \begin{bmatrix} \dot{p}_o \\ \omega_o \end{bmatrix} = \begin{bmatrix} J_{Po} \\ J_{Oo} \end{bmatrix} \dot{\theta}_a = J_o \, \dot{\theta}_a \,, \tag{2.30}$$

It is important to emphasize that the presented methodology for finding the differential kinematic equations of the object is valid under the assumption of firm contact between every finger and the object. In other words, (2.29) is valid if we can guarantee that the contact will be held for all fingers. This fact is directly related to w, the vector of angular velocities of the *virtual joints* in a contact point. If some

k-th fingertip is not in contact with the surface of the object, than it is not possible to parameterize the velocity vector v_{ck} according to (2.23). In the next section, an useful and simple introduction to a method to ensure the validity of (2.29) is presented.

2.5 Relative Kinematics

The relative kinematics of a robot hand is the mathematical description of the differences between the positions of two nearby fingers. It is of mainly importance to develop a mathematical description of such relations, since the prehension of a grasped object is directly related to the relative positions among the fingers that are in contact with it. Once again, suppose that the considered robot hand has n fingers. First, define the vector of relative positions as:

$$p_r := \begin{bmatrix} p_1 - p_2 \\ p_2 - p_3 \\ \vdots \\ p_{n-1} - p_n \end{bmatrix} \in \mathbb{R}^{3(n-1)} . \qquad (2.31)$$

Also, define the vector of finger positions as:

$$p_h := \begin{bmatrix} p_1^{\mathsf{T}} & p_2^{\mathsf{T}} & \dots & p_n^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^{3n} \qquad (2.32)$$

The relation between these two vectors is given by:

$$\underbrace{\begin{bmatrix} p_1 - p_2 \\ p_2 - p_3 \\ \vdots \\ p_{n-1} - p_n \end{bmatrix}}_{p_r} = \underbrace{\begin{bmatrix} I_3 & -I_3 & 0 & \dots & 0 & 0 \\ 0 & I_3 & -I_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & I_3 & -I_3 \end{bmatrix}}_{\tilde{A}_p} \underbrace{\begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_{n-1} \\ p_n \end{bmatrix}}_{p_h}$$

or

$$p_r = \tilde{A}_p \, p_h \qquad (2.33)$$

where $I_3 \in \mathbb{R}^{3\times 3}$ is the identity matrix and $\tilde{A}_p \in \mathbb{R}^{3(n-1)\times 3n}$. From the direct kinematics map of the k-th finger, represented by equation (2.1), we can differentiate it to obtain the differential kinematics equation in position:

$$\dot{p}_k = J^k_{Pf}\,\dot{\theta}_k\,, \tag{2.34}$$

where $J^k_{Pf} \in \mathbb{R}^{3\times n_k}$ is the position part of the finger jacobian matrix J^k_f, introduced earlier in equation (2.4).

Now, proceeding with the same methodology used to derive equation (2.18) and considering the fact that all finger joints of (Fig. 2.1) are passive ($\theta_k = \theta_{kp}$), we get:

$$\dot{p}_k = J^k_{Pf}\, J^k_C\, \dot{\theta}_a\,, \tag{2.35}$$

which is the differential kinematic equation for the linear velocity vector of the k-th finger, dependent only of the corresponding active joint velocities.

Again, similarly to (2.19), we can stack these velocities together:

$$\underbrace{\begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \\ \vdots \\ \dot{p}_n \end{bmatrix}}_{\dot{p}_h} = \underbrace{\begin{bmatrix} J^1_{Pf} & 0 & \dots & 0 \\ 0 & J^2_{Pf} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & J^n_{Pf} \end{bmatrix}}_{J_{Pf}} \underbrace{\begin{bmatrix} J^1_C \\ J^2_C \\ \vdots \\ J^n_C \end{bmatrix}}_{J_c} \dot{\theta}_a\,.$$

Note that the left-hand side of this equation is the time derivative of the previously defined vector p_h. The first block-diagonal matrix in the right-hand side is built in a similar way to J_f, but written in terms of J^k_{Pf} instead of J^k_f. Written in shorthand notation, the differential kinematics equation above is:

$$\dot{p}_h = \underbrace{J_{Pf}\, J_c}_{J_{Ph}}\, \dot{\theta}_a\,. \tag{2.36}$$

where $J_{Ph} \in \mathbb{R}^{3n\times n_m}$ is the position part of the hand Jacobian matrix J_h, written as the product of J_{Pf} and J_c. This matrix can also be written in terms of $J^k_{PF} \in \mathbb{R}^{3\times n_m}$, the position parts of the complete finger Jacobian matrices J^k_F:

$$J_{Ph} = \begin{bmatrix} {J^1_{PF}}^{\mathsf{T}} & {J^2_{PF}}^{\mathsf{T}} & \dots & {J^n_{PF}}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}\,. \tag{2.37}$$

Now, we can use the time derivative of relation (2.33) to express the relative velocities of the fingers in terms of the active joint velocities of the robot hand:

$$\dot{p}_r = \underbrace{\tilde{A}_p\, J_{Ph}}_{J_r}\, \dot{\theta}_a\,, \tag{2.38}$$

which is the relative differential kinematics of the robot hand. The matrix $J_r \in \mathbb{R}^{3n\times n_m}$ is known as the relative Jacobian matrix of the robot hand.

2.6 Modeling of the 3-Finger Adaptive Robot Hand

In this section, we develop a kinematic model for the 3-Finger Adaptive Robot Hand, from Robotiq. We cover the kinematic models for the fingers, the kinematic restrictions on the fingers velocities, details about the grasping kinematics equations for this particular robot hand and also the relative kinematics equation. These models are essential for the future development of visual servoing control schemes, since we need to find the relation between the active joint space and the operational space of the robot system before actually finding the relation between the operational space and the image space, which will be the subject of the next chapter.

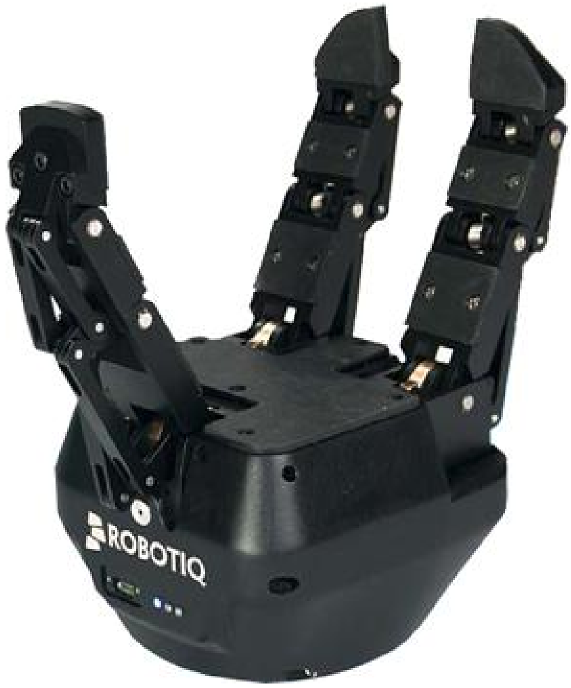


Figure 2.2: Robotiq gripper.

The Robotiq gripper is a "mechanically intelligent" gripper, capable of adapting its structure to perform adaptive grasping. Its has four internal motors, three for opening / closing each finger, and one more to perform an "scissor" movement between fingers 2 and 3. This extra motor actually couples these two fingers; one cannot move laterally without moving the other, due to the presence of an internal gear coupling between them. But they can open / close freely, each one by its own motor.

2.6.1 Forward and Differential Kinematics

Figure 2.1 in Section 2.1 illustrates the disposal of the three fingers of the 3-Finger Adaptive Robot Hand (Fig. 2.2) around the palm, as well as the convention for its links and joints. The Robotiq hand has three fingers ($n = 3$), finger 1 with $n_1 = 3$ passive joints and fingers 2 and 3 with $n_2 = n_3 = 4$ passive joints. The $n_m = 4$ active joints are located inside the palm, driving the motion of the finger joints of Fig. 2.1 by means of a transmission mechanism composed of rigid links, springs and internal gears.

For space saving, the next expressions are according to the following abbreviations and conventions:

$$\begin{aligned}
s_1^k &= \sin(\theta_{k1}) && , \quad c_1^k = \cos(\theta_{k1})\,, \\
s_{12}^k &= \sin(\theta_{k1} + \theta_{k2}) && , \quad c_{12}^k = \cos(\theta_{k1} + \theta_{k2})\,, \\
s_{123}^k &= \sin(\theta_{k1} + \theta_{k2} + \theta_{k3}) && , \quad c_{123}^k = \cos(\theta_{k1} + \theta_{k2} + \theta_{k3})\,, \\
s_4^k &= \sin(\theta_{k4}) && , \quad c_4^k = \cos(\theta_{k4})\,, \\
l_s^k &= (l_1\, s_1^k + l_2\, s_{12}^k + l_3\, s_{123}^k) && , \quad l_c^k = (l_1\, c_1^k + l_2\, c_{12}^k + l_3\, c_{123}^k) \\
\theta_{123}^k &= \theta_{k1} + \theta_{k2} + \theta_{k3} && , \quad k = 1, 2, 3\,.
\end{aligned}$$

Note that the k *superscript* is simply the index of the corresponding finger. The vectors $p_{k_{offset}} = \begin{bmatrix} p_{k_x} & p_{k_y} & p_{k_z} \end{bmatrix}$ $(k = 1, 2, 3)$, are position vectors from the origin of the palm frame $\mathcal{F}_p$ to the k-th finger base frame $\mathcal{F}_{bk}$. The vectors $p_{k_{link}} (k = 1, 2, 3)$ are the position vectors from the base of finger k (origin of $\mathcal{F}_{bk}$) to the fingertip frame $\mathcal{F}_k$, which are configuration dependent. All finger positions with respect to the palm frame $\mathcal{F}_p$ can be written as $p_k = p_{k_{offset}} + p_{k_{link}}$. The rotation matrices $R_k \in SO(3)$ are written in terms of elementary rotation matrices $R_x\,, R_y\,, R_z$. According to the conventions above and Fig. 2.1, the forward kinematics map for each finger is:

$$\begin{aligned}
p_{1_{link}} &= \begin{bmatrix} 0 & l_s^1 & l_c^1 \end{bmatrix}^{\mathbf{T}} && , \quad R_1 = R_x(-\theta_{123}^1), \\
p_{2_{link}} &= \begin{bmatrix} (l_c^2 + l_0)\, s_4^2 & -l_s^2 & (l_c^2 + l_0)\, c_4^2 \end{bmatrix}^{\mathbf{T}} && , \quad R_2 = R_y(\theta_{24}) R_x(\theta_{123}^2), \\
p_{3_{link}} &= \begin{bmatrix} -(l_c^3 + l_0)\, s_4^3 & -l_s^3 & -(l_c^3 + l_0)\, c_4^3 \end{bmatrix}^{\mathbf{T}} && , \quad R_3 = R_y(-\theta_{34})\, R_x(\theta_{123}^3)\,. \quad (2.39)
\end{aligned}$$

Next, according to Fig. 2.1, we derive the differential kinematics equations for each finger (SICILIANO *et al.*, 2009). These equations represent a linear transformation between the joint velocities and the velocity of the corresponding fingertip frame $\mathcal{F}_k$. The matrices of the transformation are called finger Jacobian matrices, and their components are dependent on the joint angles.

The analysis of the Jacobian matrices is of great importance in robotics, revealing fundamental aspects about the robot mechanism, such as its *manipulability*, singular configurations and the dynamic relation between the joint forces and torques and the forces applied in the end-effector of the robot.

According to fig. 2.1, the vectors of finger angles θ_k for $k = 1, 2, 3$ are:

$$\begin{aligned}
\theta_1 &= \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \end{bmatrix}^{\mathsf{T}} , \\
\theta_2 &= \begin{bmatrix} \theta_{21} & \theta_{22} & \theta_{23} & \theta_{24} \end{bmatrix}^{\mathsf{T}} , \\
\theta_3 &= \begin{bmatrix} \theta_{31} & \theta_{32} & \theta_{33} & \theta_{34} \end{bmatrix}^{\mathsf{T}} .
\end{aligned} \tag{2.40}$$

Thus, the differential kinematics equations for the three fingers of the Robotiq gripper are:

$$v_1 = J_f^1 \, \dot{\theta}_1 \, , \qquad v_2 = J_f^2 \, \dot{\theta}_2 \, , \qquad v_3 = J_f^3 \, \dot{\theta}_3 \, , \tag{2.41}$$

where the corresponding finger Jacobian matrices are:

$$J_f^1 = \begin{bmatrix} 0 & 0 & 0 \\ l_c^1 & d_c^1 & l_3 \, c_{123}^1 \\ -l_s^1 & -d_s^1 & -l_3 \, s_{123}^1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} ,$$

$$J_f^2 = \begin{bmatrix} -l_s^2 \, s_4^2 & -d_s^2 \, s_4^2 & -l_3 \, s_{123}^2 \, s_4^2 & (l_c^2 + l_0) \, c_4^2 \\ -l_c^2 & -d_c^2 & -l_3 \, c_{123}^2 & 0 \\ -l_s^2 \, c_4^2 & -d_s^2 \, c_4^2 & -l_3 \, s_{123}^2 \, c_4^2 & -(l_c^2 + l_0) \, s_4^2 \\ c_4^2 & c_4^2 & c_4^2 & 0 \\ 0 & 0 & 0 & 1 \\ -s_4^2 & -s_4^2 & -s_4^2 & 0 \end{bmatrix} ,$$

$$J_f^3 = \begin{bmatrix} l_s^3\, s_4^3 & d_s^3\, s_4^3 & l_3\, s_{123}^3\, s_4^3 & -(l_c^3 + l_0)\, c_4^3 \\ -l_c^3 & -d_c^3 & -l_3\, c_{123}^3 & 0 \\ -l_s^3\, c_4^3 & -d_s^3\, c_4^3 & -l_3\, s_{123}^3\, c_4^3 & -(l_c^3 + l_0)\, s_4^3 \\ c_4^3 & c_4^3 & c_4^3 & 0 \\ 0 & 0 & 0 & 1 \\ s_4^3 & s_4^3 & s_4^3 & 0 \end{bmatrix}, \tag{2.42}$$

with $d_c^k = (l_2\, c_{12}^k + l_3\, c_{123}^k)$ and $d_s^k = (l_2\, s_{12}^k + l_3\, s_{123}^k)$.

2.6.2 Internal Kinematic Constraints

Now, we proceed to derive the expressions for the kinematic constraints of each finger. Since our main objective is to perform manipulation tasks with the Robotiq gripper, it is important to identify the types of grasp in which the object is *manipulable*. Observing Fig. 2.3, we notice that the object is actually manipulable by the Robotiq hand only during a *fingertip grasp*, since it forms a "force closured grasp", but not a "force closured" grasp (to a better understanding about the principles behind *form closure grasps* and *force closure grasps*, see (PRATTICHIZZO e TRINKLE, 2008) and (MURRAY *et al.*, 1994)).

In Fig. 2.4, we can observe the finger structure in more detail. There is a relatively complex spring mechanism in the back of each finger, which is responsible for adapting the finger configuration to external forces acting in the phalanges. In practice, the overall effect is to automatically adapt the structure of the fingers to fit the shape of a grasped object during encompassing grips. However, since a grasped object is manipulable only under fingertip grips, it is not needed to find out how the fingers move when contact forces arise on phalanges other than the fingertip ones. During any feasible manipulation, the object will be in contact only with three or two fingertips, and thus, the movement of the fingers can be completely defined by the forces generated by the motors and the springs in the mechanism. In other words, during object manipulation, the fingers move in the exact same way than during free motion.

Because of this fact, we can develop a simple model for the relation between the motion of the internal motors of the hand (or, its active joints) and the motions of the finger angles showed in Fig. 2.1 (considered to be passive joints). During the free

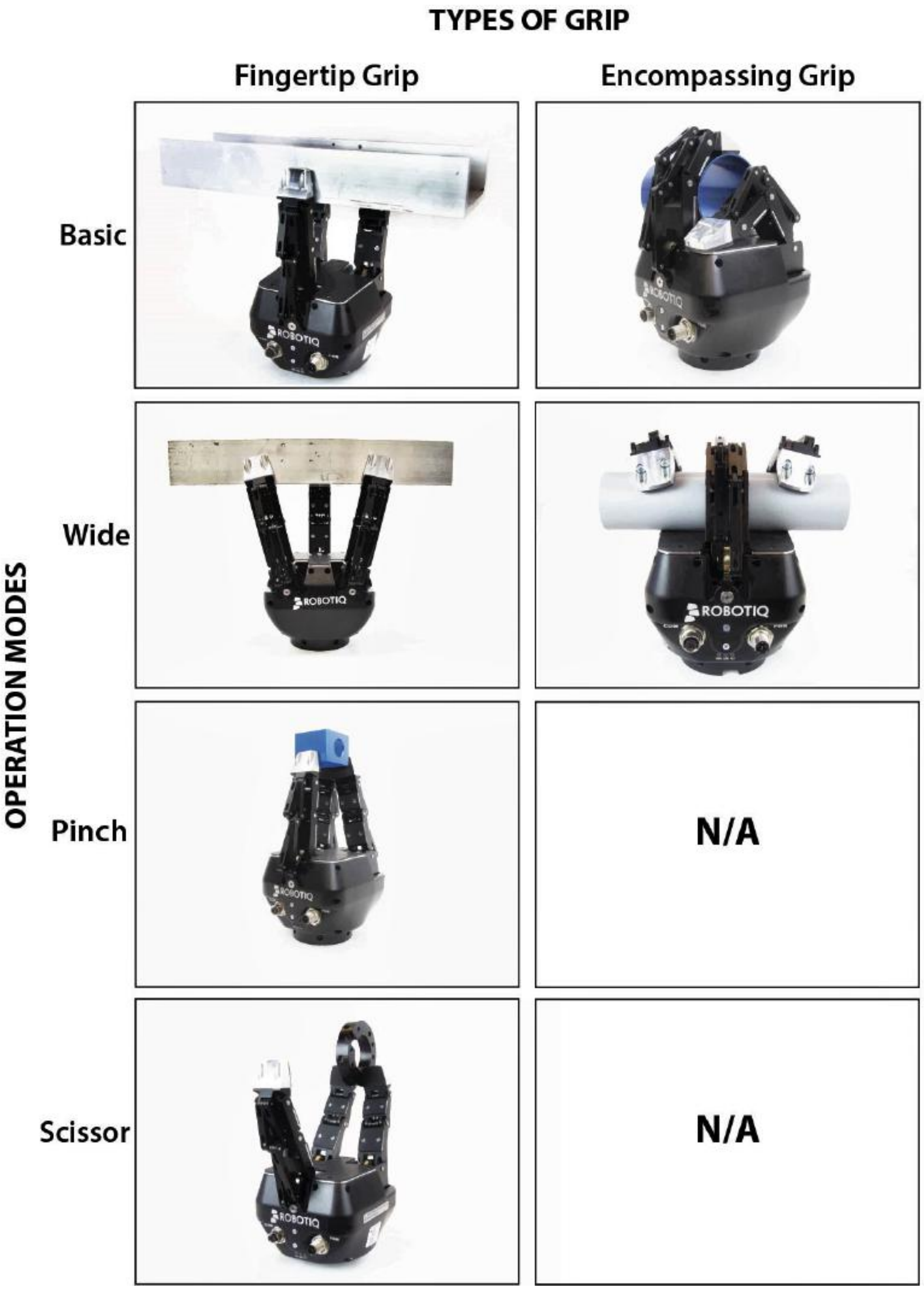


Figure 2.3: Types of grip versus operation modes of the Robotiq gripper.

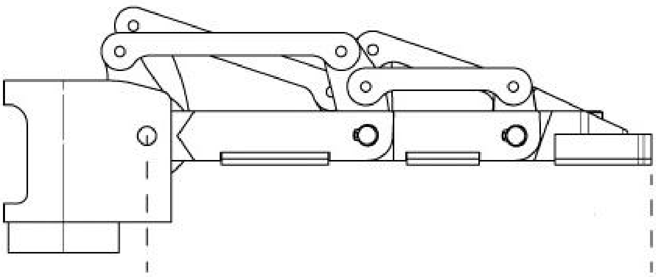


Figure 2.4: Robotiq finger, with the back-spring mechanism in detail.

motion, the motor torque tend to transfer angular momentum to the first phalange only, before it reaches a mechanical limit. Then, the movement of the first phalange stops and the back-spring mechanism forces the motor shaft to transfer momentum to the second phalange. This behavior strongly suggests that discontinuous functions can be used to model such relations.

After some observational work, we propose the following empirical relations between the finger angles θ_{kj} and the three *uncoupled* motor angles $\theta_{ka}, k = 1, 2, 3$:

$$\begin{aligned}
\theta_{k1} &= \begin{cases} \alpha_1\,\theta_{ka} + \beta_1\,, & \theta_{ka} \leq \gamma_1\,, \\ \eta_1\,, & \theta_{ka} > \gamma_1\,, \end{cases} \\
\theta_{k2} &= \begin{cases} \eta_2\,, & \theta_{ka} \leq \gamma_2\,, \\ \alpha_2\,\theta_{ka} + \beta_2\,, & \theta_{ka} > \gamma_2\,, \end{cases} \\
\theta_{k3} &= \begin{cases} \alpha_3\,\theta_{k1} + \beta_3\,, & \theta_{ka} \leq \gamma_3\,, \\ \eta_3\,, & \theta_{ka} > \gamma_3\,, \end{cases}
\end{aligned} \tag{2.43}$$

where α_k, β_k, η_k, γ_k, $k = 1, 2, 3$ are twelve constants to be determined. In addition, because of the inherent coupling between the fingers 2 and 3 in the scissor mode, we have:

$$\theta_{24} = \theta_{4a}\,, \qquad \theta_{34} = -\theta_{4a}\,. \tag{2.44}$$

where the θ_{4a} is the angle of the scissor motor. Equations (2.43) and (2.44) represent the $c_k(.)$ functions introduced in section 2.3, according to the notation for the passive joints of each finger, introduced in (2.40).

The motivation for this particular form for the empirical equations is based on extensive observation of the finger motion. For now, we are going to assume that this model is valid for some particular set of constants α_k, β_k, η_k, γ_k, $k = 1, 2, 3$.

Now, we can take the time-derivative of equations (2.43), obtaining the kinematic relationship in terms of the angular velocities:

$$\begin{aligned}
\dot\theta_{k1} &= \begin{cases} \alpha_1\,\dot\theta_{ka}\,, & \theta_{ka} \leq \gamma_1\,, \\ 0\,, & \theta_{ka} > \gamma_1\,, \end{cases} \\
\dot\theta_{k2} &= \begin{cases} 0\,, & \theta_{ka} \leq \gamma_2\,, \\ \alpha_2\,\dot\theta_{ka}\,, & \theta_{ka} > \gamma_2\,, \end{cases} \\
\dot\theta_{k3} &= \begin{cases} \alpha_3\,\dot\theta_{k1}\,, & \theta_{ka} \leq \gamma_3\,, \\ 0\,, & \theta_{ka} > \gamma_3\,. \end{cases}
\end{aligned} \tag{2.45}$$

Similarly, taking the time-derivative of (2.44):

$$\dot{\theta}_{24} = \dot{\theta}_{4a}\,, \qquad \dot{\theta}_{34} = -\dot{\theta}_{4a}\,, \tag{2.46}$$

Then, defining a modified Kronecker delta operator

$$\delta_{kj} := \begin{cases} 1\,, & \theta_{ka} \le \theta_j^*\,, \\ 0\,, & \theta_{ka} > \theta_j^*\,, \end{cases} \tag{2.47}$$

it is possible to rewrite equations (2.45) as:

$$\begin{aligned} \dot{\theta}_{k1} &= \delta_{k1}\,\alpha_1\,\dot{\theta}_{ka}\,, \\ \dot{\theta}_{k2} &= (1-\delta_{k2})\,\alpha_2\,\dot{\theta}_{ka}\,, \\ \dot{\theta}_{k3} &= \delta_{k1}\,\delta_{k3}\,\alpha_1\,\alpha_3\,\dot{\theta}_{ka}\,, \end{aligned} \tag{2.48}$$

where θ_j^* for $j=1,2,3$ is a bound for the angular position of the j-th joint for the k-th finger, such that, $\theta_j^* = \gamma_j$.

Now, consider the particular case of the Robotiq gripper, where the same actuator is shared by more than one finger. [1] The vector of active joints of the robot hand is $\theta_a = [\,\theta_{1a}\ \ \theta_{2a}\ \ \theta_{3a}\ \ \theta_{4a}\,]^\mathsf{T}$, where θ_{ka} ($k = 1,2,3$) represents the motor shaft angle for closing / opening the k-th robot finger and θ_{4a} is the angle of the scissor motor. We can split θ_a into its *unshared* and *shared* terms, as $\theta_a := [\,\theta_u^\mathsf{T}\ \ \theta_s^\mathsf{T}\,]^\mathsf{T}$, where $\theta_u = [\,\theta_{1a}\ \ \theta_{2a}\ \ \theta_{3a}\,]^\mathsf{T}$ and $\theta_s = [\,\theta_{4a}\,]$, allowing us to rewrite the velocity constraint (2.11), according to:

$$\dot{\theta}_k = \underbrace{\begin{bmatrix} J_u^k(\theta_a) & J_s^k(\theta_a) \end{bmatrix}}_{J_C^k} \underbrace{\begin{bmatrix} \dot{\theta}_u \\ \dot{\theta}_s \end{bmatrix}}_{\dot{\theta}_a}\,, \tag{2.49}$$

where J_u^k and J_s^k are the unshared and shared parts of the complete constraint Jacobian matrix J_C^k, and $\dot{\theta}_u$, $\dot{\theta}_s$ are the unshared and shared active joint velocities of the hand.

Therefore, using (2.48) and (2.46), the constraint equations for the fingers of the Robotiq gripper can be found to be:

$$\dot{\theta}_1 = \underbrace{[\,J_u^1\ \ J_s^1\,]}_{J_C^1}\,\dot{\theta}_a\,, \qquad \dot{\theta}_2 = \underbrace{[\,J_u^2\ \ J_s^2\,]}_{J_C^2}\,\dot{\theta}_a\,, \qquad \dot{\theta}_3 = \underbrace{[\,J_u^3\ \ J_s^3\,]}_{J_C^3}\,\dot{\theta}_a\,, \tag{2.50}$$

[1] This is actually not a restricted case. Other robot grippers, such as the Schunk 3-Finger Robot Hand, also have shared Dofs between two or more fingers.

and the constraint Jacobian matrices in (2.49) take the form

$$J_u^1 = \begin{bmatrix} \delta_{11}\,\alpha_1 & 0 & 0 \\ (1-\delta_{12})\,\alpha_2 & 0 & 0 \\ \delta_{11}\delta_{13}\alpha_1\alpha_3 & 0 & 0 \end{bmatrix}, \qquad J_s^1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

$$J_u^2 = \begin{bmatrix} 0 & \delta_{21}\,\alpha_1 & 0 \\ 0 & (1-\delta_{22})\,\alpha_2 & 0 \\ 0 & \delta_{21}\delta_{23}\alpha_1\alpha_3 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad J_s^2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

$$J_u^3 = \begin{bmatrix} 0 & 0 & \delta_{31}\,\alpha_1 \\ 0 & 0 & (1-\delta_{32})\,\alpha_2 \\ 0 & 0 & \delta_{31}\delta_{33}\alpha_1\alpha_3 \\ 0 & 0 & 0 \end{bmatrix}, \qquad J_s^3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{bmatrix}. \tag{2.51}$$

Thus, the complete constraint Jacobian matrices for the Robotiq hand are obtained by simply combining the equations above.

2.6.3 Grasping Kinematics

Finally, since Proposition 1 is true in the case of the Robotiq gripper, equations (2.18), $k = 1, 2, 3$ are:

$$v_1 = \underbrace{J_f^1\,J_C^1}_{J_F^1}\,\dot{\theta}_a\,, \qquad v_2 = \underbrace{J_f^2\,J_C^2}_{J_F^2}\,\dot{\theta}_a\,, \qquad v_3 = \underbrace{J_f^3\,J_C^3}_{J_F^3}\,\dot{\theta}_a\,, \tag{2.52}$$

where the augmented finger Jacobian matrices are:

$$J_F^1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \delta_{11}\,\alpha_1\,l_c^1 + (1-\delta_{12})\,\alpha_2\,d_c^1 + \delta_{11}\,\delta_{13}\,\alpha_1\,\alpha_3\,l_3\,c_{123}^1 & 0 & 0 & 0 \\ -\delta_{11}\,\alpha_1\,l_c^1 - (1-\delta_{12})\,\alpha_2\,d_s^1 - \delta_{11}\,\delta_{13}\,\alpha_1\,\alpha_3\,l_3\,s_{123}^1 & 0 & 0 & 0 \\ \delta_{11}\,\alpha_1 + (1-\delta_{12})\,\alpha_2 + \delta_{11}\,\delta_{13}\,\alpha_1\,\alpha_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$J_F^2 = \begin{bmatrix} 0 & -\delta_{21}\,\alpha_1\,l_s^2\,s_4^2 - (1-\delta_{22})\,\alpha_2\,d_s^2\,s_4^2 - \delta_{21}\,\delta_{23}\,\alpha_1\,\alpha_3\,l_3\,s_{123}^2\,s_4^2 & 0 & (l_c^2 + l_0)\,c_4^2 \\ 0 & -\delta_{21}\,\alpha_1\,l_c^2 - (1-\delta_{22})\,\alpha_2\,d_c^2 - \delta_{21}\,\delta_{23}\,\alpha_1\,\alpha_3\,l_3\,c_{123}^2 & 0 & 0 \\ 0 & -\delta_{21}\,\alpha_1\,l_s^2\,c_4^2 - (1-\delta_{22})\,\alpha_2\,d_s^2\,c_4^2 - \delta_{21}\,\delta_{23}\,\alpha_1\,\alpha_3\,l_3\,s_{123}^2\,c_4^2 & 0 & -(l_c^2 + l_0)\,s_4^2 \\ 0 & \delta_{21}\,\alpha_1\,c_4^2 + (1-\delta_{22})\,\alpha_2\,c_4^2 + \delta_{21}\,\delta_{23}\,\alpha_1\,\alpha_3\,c_4^2 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\delta_{21}\,\alpha_1\,s_4^2 - (1-\delta_{22})\,\alpha_2\,s_4^2 - \delta_{21}\,\delta_{23}\,\alpha_1\,\alpha_3\,s_4^2 & 0 & 0 \end{bmatrix},$$

$$J_F^3 = \begin{bmatrix} 0 & 0 & \delta_{31}\,\alpha_1\,l_s^3\,s_4^3 + (1-\delta_{32})\,\alpha_2\,d_s^3\,s_4^3 + \delta_{31}\,\delta_{33}\,\alpha_1\,\alpha_3\,l_3\,s_{123}^3\,s_4^3 & -(l_c^3 + l_0)\,c_4^3 \\ 0 & 0 & -\delta_{31}\,\alpha_1\,l_c^3 - (1-\delta_{32})\,\alpha_2\,d_c^3 - \delta_{31}\,\delta_{33}\,\alpha_1\,\alpha_3\,l_3\,c_{123}^3 & 0 \\ 0 & 0 & -\delta_{31}\,\alpha_1\,l_s^3\,c_4^3 - (1-\delta_{32})\,\alpha_2\,d_s^3\,c_4^3 - \delta_{31}\,\delta_{33}\,\alpha_1\,\alpha_3\,l_3\,s_{123}^3\,c_4^3 & (l_c^3 + l_0)\,s_4^3 \\ 0 & 0 & \delta_{31}\,\alpha_1\,c_4^3 + (1-\delta_{32})\,\alpha_2\,c_4^3 + \delta_{31}\,\delta_{33}\,\alpha_1\,\alpha_3\,c_4^3 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & \delta_{31}\,\alpha_1\,s_4^3 + (1-\delta_{32})\,\alpha_2\,s_4^3 + \delta_{31}\,\delta_{33}\,\alpha_1\,\alpha_3\,s_4^3 & 0 \end{bmatrix}.$$

From this point, we can use expression (2.21) to calculate the *hand Jacobian matrix* J_h by simply stacking the matrices above:

$$J_h = \begin{bmatrix} J_F^1 \\ J_F^2 \\ J_F^3 \end{bmatrix}. \tag{2.53}$$

In order to complete the description of the grasping kinematics for the Robotiq hand, we need a model for the manipulated object, expressed by the the Adjoint matrices A_1, A_2, A_3. These matrices are dependent on the geometry of the object, and therefore, they are going to be discussed in chapter 4, together with details about the contact matrices H_k.

2.6.4 Relative Kinematics

Because the Robotiq Gripper has three fingers, the relative position vector is:

$$p_r = \begin{bmatrix} p_1 - p_2 \\ p_2 - p_3 \end{bmatrix},$$

and the vector of finger positions is $p_h = [\,p_1^\mathsf{T}\ \ p_2^\mathsf{T}\ \ p_3^\mathsf{T}\,]^\mathsf{T}$. The relation between them is given by (2.33):

$$p_r = \tilde{A}_p\,p_h\,, \qquad \tilde{A}_p = \begin{bmatrix} I_3 & -I_3 & 0 \\ 0 & I_3 & -I_3 \end{bmatrix},$$

To find the expression for J_{Ph}, we use equation (2.37) and the position part of the augmented finger Jacobian matrices, $J^k_{F_P}$ $(k = 1, 2, 3)$, obtaining:

$$J_{Ph} = \begin{bmatrix} J^1_{PF} \\ J^2_{PF} \\ J^3_{PF} \end{bmatrix}, \tag{2.54}$$

with

$$J^1_{PF} \;=\; \begin{bmatrix} 0 & 0 & 0 & 0 \\ \delta_{11}\,\alpha_1\,l^1_c + (1-\delta_{12})\,\alpha_2\,d^1_c + \delta_{11}\,\delta_{13}\,\alpha_1\,\alpha_3\,l_3\,c^1_{123} & 0 & 0 & 0 \\ -\delta_{11}\,\alpha_1\,l^1_c - (1-\delta_{12})\,\alpha_2\,d^1_s - \delta_{11}\,\delta_{13}\,\alpha_1\,\alpha_3\,l_3\,s^1_{123} & 0 & 0 & 0 \end{bmatrix},$$

$$J^2_{PF} \;=\; \begin{bmatrix} 0 & -\delta_{21}\,\alpha_1\,l^2_s\,s^2_4 - (1-\delta_{22})\,\alpha_2\,d^2_s\,s^2_4 - \delta_{21}\,\delta_{23}\,\alpha_1\,\alpha_3\,l_3\,s^2_{123}\,s^2_4 & 0 & (l^2_c + l_0)\,c^2_4 \\ 0 & -\delta_{21}\,\alpha_1\,l^2_c - (1-\delta_{22})\,\alpha_2\,d^2_c - \delta_{21}\,\delta_{23}\,\alpha_1\,\alpha_3\,l_3\,c^2_{123} & 0 & 0 \\ 0 & -\delta_{21}\,\alpha_1\,l^2_s\,c^2_4 - (1-\delta_{22})\,\alpha_2\,d^2_s\,c^2_4 - \delta_{21}\,\delta_{23}\,\alpha_1\,\alpha_3\,l_3\,s^2_{123}\,c^2_4 & 0 & -(l^2_c + l_0)\,s^2_4 \end{bmatrix},$$

$$J^3_{PF} \;=\; \begin{bmatrix} 0 & 0 & \delta_{31}\,\alpha_1\,l^3_s\,s^3_4 + (1-\delta_{32})\,\alpha_2\,d^3_s\,s^3_4 + \delta_{31}\,\delta_{33}\,\alpha_1\,\alpha_3\,l_3\,s^3_{123}\,s^3_4 & -(l^3_c + l_0)\,c^3_4 \\ 0 & 0 & -\delta_{31}\,\alpha_1\,l^3_c - (1-\delta_{32})\,\alpha_2\,d^3_c - \delta_{31}\,\delta_{33}\,\alpha_1\,\alpha_3\,l_3\,c^3_{123} & 0 \\ 0 & 0 & -\delta_{31}\,\alpha_1\,l^3_s\,c^3_4 - (1-\delta_{32})\,\alpha_2\,d^3_s\,c^3_4 - \delta_{31}\,\delta_{33}\,\alpha_1\,\alpha_3\,l_3\,s^3_{123}\,c^3_4 & (l^3_c + l_0)\,s^3_4 \end{bmatrix}.$$

Chapter 3

Visual Servoing and Control Design

In this chapter, we cover: the derivation of a simple camera model, widely used in visual servoing schemes; the Cartesian-to-image coordinate transformations, necessary for control in the image space; two different approaches to visual servoing control, known as *image-based visual servoing* or *IBVS*, and *hybrid visual servoing* or *HVS* and five control algorithms for the robot hand, combining the methodologies discussed previously: (i) independent finger control, (ii) prehension control, (iii) object control, (iv) IBVS control and (v) HVS control.

3.1 Camera Models and Depth Recovery

In this section, a simple, but suitable camera model is described. A *pinhole camera model* simply represents a relation between the 3D coordinates of a generic point on the Cartesian space and its projection onto the image plane.

Figure (3.1) shows the schematic of the visual servoing system. Let $p_c = [\ x_c \quad y_c \quad z_c\ ]^\mathsf{T}$ be the position vector of the object frame $\mathcal{F}_o$ with respect to the camera frame $\mathcal{F}_c$, and let $p_v = [\ x_v \quad y_v\ ]^\mathsf{T}$ be the position vector of the object frame $\mathcal{F}_o$ with respect to the image frame $\mathcal{F}_v$. From the frontal perspective projection model of a pinhole camera (CHAUMETTE e HUTCHINSON, 2008), the Cartesian

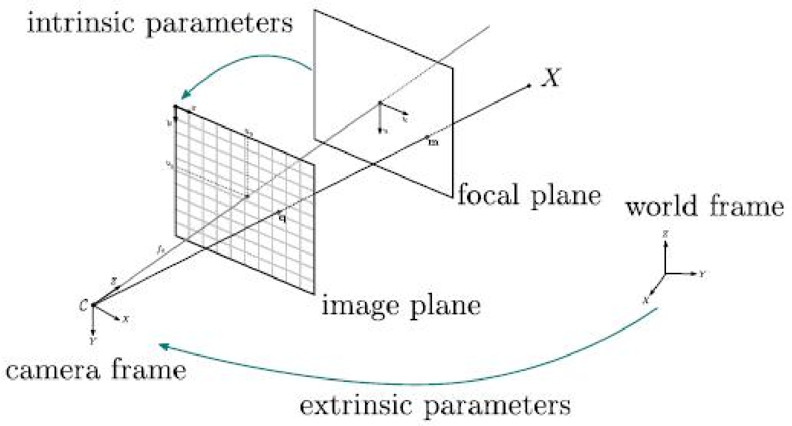


Figure 3.1: Perspective transformation.

space can be related to the image space as:

$$\begin{bmatrix} x_v \\ y_v \end{bmatrix} = \frac{f}{z_c} \begin{bmatrix} \alpha_x & 0 \\ 0 & \alpha_y \end{bmatrix} \begin{bmatrix} x_c \\ y_c \end{bmatrix} + \begin{bmatrix} x_{v0} \\ y_{v0} \end{bmatrix}, \tag{3.1}$$

where f is the focal length of camera lens, α_x, α_y are the camera scaling factors due to the finite dimensions of the image sensor and x_{v0}, y_{v0} are offsets values that adjust the position of the origin of the pixel coordinate system with respect to the optical axis.

Defining the augmented vector $\tilde{p}_v = [\, x_v \;\; y_v \;\; 1\,]^\mathsf{T}$, the nonlinear Cartesian-to-image coordinate transformation (3.1) can be written in a linear matrix form:

$$z_c\,\tilde{p}_v = \Omega\, p_c\,, \qquad \Omega = \begin{bmatrix} f\,\alpha_x & 0 & x_{v0} \\ 0 & f\,\alpha_y & y_{v0} \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.2}$$

where $z_c \in \mathbb{R}$ is the depth coordinate expressed in the camera frame $\mathcal{F}_c$ and $\Omega \in \mathbb{R}^{3\times 3}$ is the so-called matrix of intrinsic parameters of the camera. A number of calibration techniques for off-line estimation of these parameters can be found in (CHAUMETTE e HUTCHINSON, 2008).

Analyzing the Cartesian-to-image coordinate transformation (3.2), we can observe that it is not possible to obtain any explicit information about the depth coordinate using only the object centroid coordinates x_v and y_v. In general, this is true for bi-dimensional images, because the depth information is lost after the the perspective transformation. In other words, we can not *explicitly* calculate the depth between the object and camera frames using only a single camera. However, there are some methods for recovering this information. (i) *directly*, using a stereo vision system or a depth sensor; (ii) *indirectly*, from the geometric model of the object or using the image projected area of the object.

In the next two subsections, we consider the *depth recovery problem* for a target object that moves freely in the Cartesian space while being monitored by a single camera. Two depth reconstruction methods, based on *planar homography* and *image projected area* are presented to recover the depth information from 2D images. Each one inspires two different visual servoing methodologies, as will be clear in the next sections.

3.1.1 Planar homography for depth recovery

Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be the frames attached to the object in two different time instants t_1 and t_2. The position vectors of the corresponding frames, denoted by $p_{o,1}$ and $p_{o,2}$, are related as:

$$p_{o,1} = R_{12}\, p_{o,2} + r_{12} \,, \tag{3.3}$$

where r_{12} and R_{12} are respectively the translational vector and the rotation matrix of frame $\mathcal{F}_2$ with respect to frame $\mathcal{F}_1$.

The planar homography matrix $\mathcal{H}$ can be used to evaluate the position displacement of the origins of the frames $\mathcal{F}_1$ and $\mathcal{F}_2$ as:

$$\mathcal{H} = R_{12} + \frac{1}{d_2}\, r_{12}\, n_2^{\mathsf{T}} \,, \qquad p_{o,1} = \mathcal{H}\, p_{o,2} \,, \tag{3.4}$$

where n_2 is the unit normal vector of the plane which contains the object expressed in the frame $\mathcal{F}_2$, and d_2 is the distance of the plane from the origin of frame $\mathcal{F}_2$. Notice that, the information obtained from the homography matrix $\mathcal{H}$ can be used to define the ratio $\rho_z := z_1/z_2$ in terms of measurable quantities of the image centroid coordinates x_v, y_v for each time instant.

Inspired by these ideas, we can define the augmented vector of the image centroid coordinates as (CHAUMETTE e HUTCHINSON, 2008):

$$\tilde{p}_{v_{\mathcal{H}}} = [\, x_{v_{\mathcal{H}}} \;\; y_{v_{\mathcal{H}}} \;\; z_{v_{\mathcal{H}}} \,]^{\mathsf{T}} = [\, x_v \;\; y_v \;\; \ln(z_c) \,]^{\mathsf{T}} \,, \tag{3.5}$$

where $\ln(\cdot)$ is the natural logarithm function and $\ln(z_c)$ is a supplementary normalized depth coordinate. Note that this definition is valid in the case of depth recovery by planar homography and also in the case of direct depth measurement by distance sensors, such as the Microsoft Kinect.

3.1.2 Image projected area for depth recovery

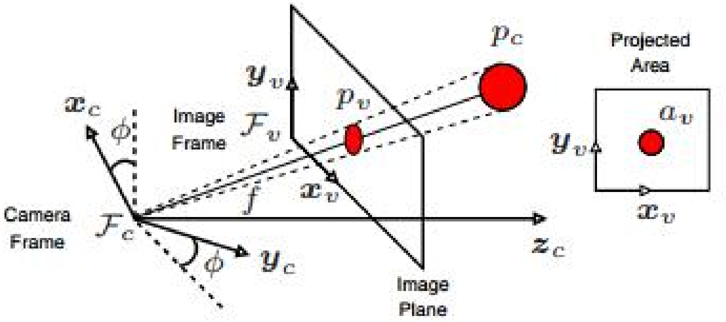


Figure 3.2: Image projected area.

Here, we consider that the depth measurement can be indirectly estimated using the image projected area of the object. The key idea is to use a target with *spherical* geometry so that the projected area in the image space becomes invariant with respect to the object rotations. Let $a_v \in \mathbb{R}^+$ be the projected area of the target object expressed in the image frame $\mathcal{F}_v$, as we can see in Fig.3.2. The dynamics of the depth-to-area transformation is given by (ZACHI *et al.*, 2004):

$$\dot{a}_v = -\left(\frac{2a_v}{z_c}\right)\dot{z}_c \,. \tag{3.6}$$

Here, the following assumptions will be considered:

(A1) The image projected area a_v is bounded and satisfies the inequality:

$$0 < a_{min} < a_v(t) < a_{max} \,;$$

(A2) The sign of z_c is assumed to be constant and known.

Under these assumptions, we can solve the separable differential equation (3.6), obtaining:

$$z_c\,(a_v)^{\frac{1}{2}} = \beta \,, \tag{3.7}$$

and we can determine the constant β by measuring the image projected area of the object at a known distance from the camera. This expression is used for mapping the image projected area of a known object into its distance z_c from the camera frame.

Note that expression (3.7) is valid not only for spherical objects, but also for objects of fixed orientation with respect to the camera frame. Changes in the object orientation can cause changes in the projected area at fixed distances, leading to inaccurate results in case of use of (3.7).

3.2 Visual-based Modeling for a Robot Hand

In this section, we present the visual-based modeling for a robot hand performing manipulation tasks with a grasped object. We limit our discussion to cover only the modeling aspects that will be useful in the control design, postponing the explicit details about the control algorithms to the next section. We assume that the depth coordinate with respect to the camera frame is measurable, directly or indirectly, according to the methodologies studied in sections 3.1.1 and 3.1.2, and also that the image centroid coordinates are computed from an image processing algorithm.

In section 3.3, we derive two different vision-based modeling schemes: (i) Hybrid or 2-1/2-D visual servoing, with direct depth measurement using a depth sensor, detailed at section 3.3.4; (ii) 3D visual servoing with indirect depth measurement using the image projected area of the object, detailed at section 3.3.5. Both schemes rely on a frame-to-frame transformation, mapping positions and velocities of the object in the palm frame to the camera frame.

Analogously to (2.4), the differential kinematics equation (2.29) can be written as:

$$v_o = \begin{bmatrix} \dot{p}_o \\ \omega_o \end{bmatrix} = \underbrace{\begin{bmatrix} J_{Po} \\ J_{Oo} \end{bmatrix}}_{J_o} \dot{\theta}_a = J_o(\theta_a)\,\dot{\theta}_a\,. \tag{3.8}$$

where v_o is the velocity of the object with respect to the palm frame $\mathcal{F}_p$, J_{Po} and J_{Oo} are the position and orientation parts of the object Jacobian matrix J_o and $\dot{\theta}_a$ is the velocity of the active joints, or motor shafts of the hand.

The position vector of the object frame $\mathcal{F}_o$ can be related to the palm and camera frames, $\mathcal{F}_p$ and $\mathcal{F}_c$, using the following coordinate transformation:

$$p_c = R_{cp}\,p_o + r_{cp}\,, \tag{3.9}$$

where $R_{cp} \in \mathbb{SO}^3$ is the rotation matrix and $r_{cp} \in \mathbb{R}^3$ is the translational vector corresponding to the coordinate transformation of the palm frame $\mathcal{F}_p$ with respect to the camera frame $\mathcal{F}_c$. It is worth mentioning that the pair (R_{cp}, r_{cp}) denotes the *extrinsic parameters* of the camera (CHAUMETTE e HUTCHINSON, 2008). Differentiating (3.9) with respect to time, we obtain:

$$\dot{p}_c = R_{cp}\,\dot{p}_o + \dot{R}_{cp}\,p_o + \dot{r}_{cp}\,. \tag{3.10}$$

Also, we can write $\omega_c \in \mathbb{R}^3$, the angular velocity of the object with respect to the camera, by means of:

$$\omega_c = R_{cp}\,\omega_o + \omega_{cp}\,, \tag{3.11}$$

where $\omega_o \in \mathbb{R}^3$ and $\omega_{cp} \in \mathbb{R}^3$ are the angular velocity of the object with respect to palm frame $\mathcal{F}_p$ and the angular velocity of the hand with respect to camera frame $\mathcal{F}_c$, respectively.

The time-derivative of the rotation matrix R_{cp} can be related to ω_{cp} by means of the skew-symmetric matrix $\mathcal{Q}(.)$ (MURRAY *et al.*, 1994):

$$\dot{R}_{cp} = \mathcal{Q}(\omega_{cp})\,R_{cp}\,. \tag{3.12}$$

Combining (3.10) and (3.11) with the position and orientation parts of the object Jacobian matrix from (3.8) and with (3.12), we obtain:

$$\dot{p}_c = R_{cp}\,J_{Po}\,\dot{\theta}_a + \mathcal{Q}(\omega_{cp})\,R_{cp}\,p_o + \dot{r}_{cp}\,, \tag{3.13}$$

$$\omega_c = R_{cp}\,J_{Oo}\,\dot{\theta}_a + \omega_{cp}\,, \tag{3.14}$$

It is worth mentioning that (3.13) and (3.14) are quite general, since they consider the existence of translation and rotation motions between the camera and the palm frames. This assumption is valid, for example, when the camera is fixed on a tripod and the robot hand is coupled to the arm tip of a large-scale manipulator performing object manipulation tasks.

From this point forward, we must find the relation between the positions and velocities of the object in the camera frame with related quantities in the image space. Since this relation is directly related to the control problem of defining a state space, it is going to be discussed in sections 3.3.4 and 3.3.5, where we define our state space in terms of quantities in the image space.

3.3 Control Design

In this section, we discuss five different approaches for controlling the Robotiq hand, from the simplest approach of controlling each robot finger independently until the visual servoing control methodologies, motivated in the previous sections.

All five control design methodologies are based on the *kinematic control* approach at the velocity level due to its simplicity of implementation and satisfactory performance when:

H1 *The task does not require too fast motions or rapid accelerations.*

H2 *The mechanism has high gear reduction ratios.*

Despite the robot hand is an electromechanical system actuated by motor torques, the kinematic control approach considers a velocity command as the system input. This assumption is valid due to the presence of a low-level control loop with high performance, which imposes any specified reference velocity to the finger joints. In this case, under assumptions (H1) - (H2), the unconsidered dynamics of the system appear on its equations as a small perturbation, which can therefore be neglected if the the decentralized control control gains are high enough (SICILIANO *et al.*, 2009). In this context, the movements of the fingers can be defined simply by

$$\dot{\theta}_a \approx u \,, \tag{3.15}$$

where $u \in \mathbb{R}^{n_m}$ is the velocity control signal applied to the motor drivers of the corresponding active joints of the hand.

3.3.1 Finger control

Here, the control goal is to make all fingers track a desired reference trajectory specified in the palm frame. From (2.2), defining the state vector

$$\xi := \begin{bmatrix} S_1\, h_1 \\ S_2\, h_2 \\ S_3\, h_3 \end{bmatrix} = \underbrace{\begin{bmatrix} S_1 & 0 & 0 \\ 0 & S_2 & 0 \\ 0 & 0 & S_3 \end{bmatrix}}_{S} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} , \tag{3.16}$$

where $S_k \in \mathbb{R}^{m_k \times 6}$, $m_k \leq 6$, $k = 1, 2, 3$, are *selection matrices*, used to choose m_k finger pose coordinates for control. This is necessary because the Robotiq hand has only four Dofs and many kinematic constraints, limiting the range of directly controllable directions for the fingers. Each line of the selection matrices is a 6-th dimensional standard basis vector $e_i = [\,0 \;\; \ldots \;\; 0 \;\; 1 \;\; 0 \;\; \ldots \;\; 0\,]$, selecting a particular direction / angle of the finger pose.

Differentiating the state vector ξ with respect to time and using (2.6):

$$\dot{\xi} = S \underbrace{\begin{bmatrix} T_{A1} & 0 & 0 \\ 0 & T_{A2} & 0 \\ 0 & 0 & T_{A3} \end{bmatrix}}_{T_A} \underbrace{\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}}_{v_h} , \tag{3.17}$$

where T_A is a block diagonal matrix of the transformation matrices introduced in section 2.2. Note that the last term in the right-hand side of (3.17) is the velocity vector of the robot hand, introduced in section 2.4. Combining (3.17) with (2.19) and considering the kinematic control approach (3.15), we obtain:

$$\dot{\xi} = J_s\, u\,, \qquad J_s = S\, T_A\, J_h\,. \tag{3.18}$$

where J_s is the system Jacobian matrix. Let $\tilde{\xi} := \xi_d(t) - \xi$ be the error state vector, expressed in terms of the generalized coordinates as follows:

$$\tilde{\xi} = S \left(\begin{bmatrix} h_{1d} \\ h_{2d} \\ h_{3d} \end{bmatrix} - \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} \right) , \tag{3.19}$$

where ξ_d is the desired value for the state vector, expressed in terms of the desired pose coordinates for each finger in the palm frame. From (3.18) and (3.19), the dynamics of the error state vector is given by:

$$\dot{\tilde{\xi}} = \dot{\xi}_d - J_s\, u\,. \tag{3.20}$$

Now, we proceed to the derivation of a suitable control law in the joint space.

Theorem 3.3.1 *Consider the control system given by (3.18), with an full-rank system Jacobian matrix 3.18. Under the assumption of kinematic control 3.15, the joint* *control signal that ensures the asymptotic convergence of the error state vector $\tilde{\xi}$ to zero is:*

$$u = J_s^{\dagger}\,(\,\dot{\xi}_d + \Lambda\,\tilde{\xi}\,)\,, \tag{3.21}$$

where $\Lambda = \Lambda^T$ is a positive-definite gain matrix and $J_s^{\dagger}$ is the Moore-Penrose pseudo-inverse of the system Jacobian matrix.

Proof 2 *See Appendix A - A.2.*

Now we analyze the particular case where the selection matrices are:

$$S_1 = S_2 = S_3 = [\,S_p \;\; 0\,]\,,$$

where $0 \in \mathbb{R}^{m\times 3}$ is a three-column null matrix and $S_p \in \mathbb{R}^{m\times 3}$ is another selection matrix, with $m <= 3$. Choosing selection matrices of this type is the equivalent of composing the state vector with only the finger positions, ignoring the orientation angles. In this case, the state vector reduces to:

$$\xi = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}. \tag{3.22}$$

Thus, using (2.37), the dynamics of the error state vector is simply

$$\dot{\tilde{\xi}} = \dot{\xi}_d - J_{Ph}\,u\,, \tag{3.23}$$

where J_{Ph} is the position part of the hand Jacobian matrix, introduced in section 2.5.

3.3.2 Prehension control

In this section, we present the prehension control for the Robotiq gripper. The prehension control is a method for controlling the relative positions of the fingers. This is important because, during finger coordinate motions, we want to ensure that at least some coordinates of the relative positions of the fingers remain constant. The active control of the relative positions can ensure that, for example, a manipulated object can be properly grasped during any motion, without being compressed by the fingers or falling over. If well designed, the prehension control scheme can ensure a *force-closure* grasp in some classes of objects, without needing any auxiliary force control scheme.

From (2.31), define the relative state vector $\xi_r = S_r\, p_r$ (for $n = 3$ fingers):

$$\xi_r := S_r\, p_r = S_r \begin{bmatrix} p_1 - p_2 \\ p_2 - p_3 \end{bmatrix} = S_r \begin{bmatrix} \Delta p_{12} \\ \Delta p_{23} \end{bmatrix} , \qquad S_r = \begin{bmatrix} S_{12} & 0 \\ 0 & S_{23} \end{bmatrix} , \tag{3.24}$$

where $S_{12} \in \mathbb{R}^{m_1 \times 3}$ and $S_{23} \in \mathbb{R}^{m_2 \times 3}$, $m_1, m_2 \leq 3$ are selection matrices for each pair of finger positions. The number of lines m_1, m_2 represents how many directions of each pair is selected for control.

This way, considering the kinematic control approach (3.15) with equation (2.38), we can write the dynamics of the relative state vector:

$$\dot{\xi}_r = S_r\, J_r\, u\, , \qquad J_r = \tilde{A}_p\, J_{Ph}\, , \tag{3.25}$$

Let $\tilde{\xi}_r := \xi_{rd}(t) - \xi_r$ be the error state vector:

$$\tilde{\xi}_r = S_r \left(\begin{bmatrix} \Delta p_{12d} \\ \Delta p_{23d} \end{bmatrix} - \begin{bmatrix} \Delta p_{12} \\ \Delta p_{23} \end{bmatrix} \right) , \tag{3.26}$$

Using (3.25) and (3.26), the dynamics of the error state vector is given by:

$$\dot{\tilde{\xi}}_r = \dot{\xi}_{rd} - S_r\, J_r\, u\, . \tag{3.27}$$

Before proceeding to the derivation of a suitable control law for the object prehension, we should make a remark about how to control the relative positions of the fingers to achieve the objective of ensuring an appropriate grasp.

Suppose that, before the execution of any manipulation task, the object is firmly grasped by the robot hand by means of some ad-hoc grasping algorithm. Then, using the forward kinematics map of the fingers, we can calculate the relative position state $\xi_r(t)$ at time $t = 0$: $\xi_r(0) = \xi_{r0}$, and set this initial state as a constant reference for the relative positions $\xi_{rd} = \xi_{r0}$. The desired relative velocities $\dot{\xi}_{rd}$ must be set to zero.

This methodology can ensure that the fingers configuration around the object remain unchanged during manipulation tasks, and therefore preventing the object to fall or to be compressed due to excessive forces, without requiring a force control algorithm. These results are summarized in the following theorem.

Theorem 3.3.2 *Consider the control system given by (3.25), with an invertible relative Jacobian matrix 2.38 and initial finger relative state 3.24 given by* $\xi_r(0)$. *Also, consider that:*

- *the object is firmly grasped by the robot hand at time* $t = 0$;
- *kinematic control assumption 3.15 is valid;*
- *constant relative state reference* ($\dot{\xi}_{rd} = 0$);

The joint control signal that ensures the asymptotic convergence of the relative error 3.26 to zero, and thus a constant relative position 2.31 for the fingers is then:

$$u = (S_r\, J_r)^{\dagger}\, (\, \Lambda_r\, \tilde{\xi}_r\,)\,, \tag{3.28}$$

where $\Lambda = \Lambda^T$ *is a positive-definite gain matrix and* $(S_r\, J_r)^{\dagger}$ *is the Moore-Penrose pseudo-inverse of the system relative Jacobian matrix 3.25.*

Proof 3 *See Appendix A - A.3.*

In the next sections, we are going to design control schemes for object manipulation tasks. The relative state vector introduced in this section is going to compose the next state vectors that are going to be defined, in order to ensure the object prehension as a secondary control objective.

3.3.3 Object manipulation control

In this section, we consider the problem of controlling the object position with respect to the palm frame $\mathcal{F}_p$, as well as the relative position of the fingers, to ensure a firm grasp during the task. We are not going to consider the problem of controlling the object orientation, because of geometrical aspects of the fingertip grip and the kinematic constraints of the Robotiq hand.

Define the state vector

$$\xi := \begin{bmatrix} \xi_o \\ \xi_r \end{bmatrix}\,, \tag{3.29}$$

where $\xi_o = S_o\, p_o$ and $\xi_r = S_r\, p_r$, combining the *object absolute position and finger relative positions* into the same state, whose controlled directions are limited by the selection matrices $S_o \in \mathbb{R}^{m_o \times 3}$ and $S_r \in \mathbb{R}^{m_r \times 3}$, $m_o, m_r \leq 3$.

From equation (2.30), (3.27) and considering the kinematic control approach (3.15), the dynamics of the object state error during manipulation tasks is given by:

$$\dot{\xi} = \begin{bmatrix} \dot{\xi}_o \\ \dot{\xi}_r \end{bmatrix} = J_s\, u\,, \qquad J_s = \begin{bmatrix} S_o\, J_{Po} \\ S_r\, J_r \end{bmatrix} u\,, \tag{3.30}$$

where $J_s \in \mathbb{R}^{(m_o+m_r)\times n_m}$ is the system Jacobian matrix. Here, we consider that the control goal for the robot hand is to follow a time-varying reference trajectory $\xi_{od}(t)$ for the object position state, while keeping a constant relative position state ξ_{rd}.

Let $\tilde{\xi} := \xi_d - \xi$ be the error state vector:

$$\tilde{\xi} := \begin{bmatrix} \tilde{\xi}_o \\ \tilde{\xi}_r \end{bmatrix} = \begin{bmatrix} \xi_{od} \\ \xi_{rd} \end{bmatrix} - \begin{bmatrix} \xi_o \\ \xi_r \end{bmatrix} . \tag{3.31}$$

From (3.31) and (3.30), the dynamics of the state error takes the form:

$$\dot{\tilde{\xi}} = \begin{bmatrix} \dot{\tilde{\xi}}_o \\ \dot{\tilde{\xi}}_r \end{bmatrix} = \begin{bmatrix} \dot{\xi}_{od} \\ \dot{\xi}_{rd} \end{bmatrix} - J_s\, u\,. \tag{3.32}$$

Under these assumptions, we proceed to the derivation of a control law in the joint space.

Theorem 3.3.3 *Consider the control system (3.30), with initial relative finger state 3.24 $\xi_r(0)$ and full-rank system Jacobian 3.30. Also, consider that:*

- *the object is firmly grasped by the robot hand at time $t = 0$;*
- *kinematic control assumption 3.15 is valid;*
- *constant relative state reference ($\dot{\xi}_{rd} = 0$);*

The joint velocity control signal that ensures the asymptotic convergence of the error state vector 3.31 to zero and therefore maintaining the relative position state constant at $\xi_r(0)$ during the whole manipulation task is:

$$u = J_s^{\dagger} \begin{bmatrix} \dot{\xi}_{od} + \Lambda_o\, \xi_o \\ \Lambda_r\, \xi_r \end{bmatrix} , \tag{3.33}$$

where $\Lambda_o = \Lambda_o^T$ and $\Lambda_r = \Lambda_r^T$ are positive definite gain matrices, and $J_s^{\dagger}$ is the Moore-Penrose pseudo-inverse of the system Jacobian.

Proof 4 *See Appendix A - A.4.*

3.3.4 Hybrid visual servoing control

In the next two subsections, we are going to discuss the position control of the grasped object in the image space, as well as the finger relative position control as secondary objective. The problem of object orientation control in the image space is not going to be discussed, since we are going to focus mostly in spherical objects, whose orientation is hard to recover by image processing algorithms.

Here, we present the control design for the hybrid visual servoing control scheme, or HVS, which combines the benefits of position-based and image-based approaches. The control error is defined in terms of both operational space and image space variables, and also the fingers relative positions.

From section 3.1.1 and 3.3.2, define the system state vector as

$$\xi := \begin{bmatrix} \xi_z \\ \xi_r \end{bmatrix} = \begin{bmatrix} S_z\, \tilde{p}_{v_{\mathcal{H}}} \\ S_r\, p_r \end{bmatrix}, \tag{3.34}$$

where $S_z \in \mathbb{R}^{m_z \times 3}$ and $S_r \in \mathbb{R}^{m_r \times 6}$, $m_z, m_r \leq 3$ are appropriate selection matrices.

Taking the time-derivative of ξ_z and using (3.1), (3.5), and (3.13), we obtain:

$$\dot{\xi}_z = S_z\, \Omega_z \left[R_{cp}\, J_{Po}\, \dot{\theta}_a + \mathcal{Q}(\omega_{cp})\, R_{cp}\, p_o + \dot{r}_{cp} \right], \tag{3.35}$$

with

$$\Omega_z(p_v, z_c) = \frac{1}{z_c} \begin{bmatrix} f\,\alpha_x & 0 & -x_v + x_{v_0} \\ 0 & f\,\alpha_y & -y_v + y_{v_0} \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.36}$$

where $\Omega_v \in \mathbb{R}^{3\times 3}$ is the well-known interaction matrix.

Note that the subsystem described by the (3.35) is very general, accounting for the relative motions of the camera frame $\mathcal{F}_c$ with respect to the palm frame $\mathcal{F}_p$. Next, considering the kinematic control approach (3.15), upposing that there is no relative motion between the camera and the robot hand, and thus $\dot{r}_{cp} = 0$ and $\omega_{cp} = 0$, yields:

$$\dot{\xi}_z = S_z\, J_z\, u\,, \qquad J_z(p_v, z_c, \theta_a) = \Omega_z\, R_{cp}\, J_{Po}\,, \tag{3.37}$$

where $J_z \in \mathbb{R}^{(m_z+m_r)\times n_m}$ is the so-called HVS image Jacobian.

Combining (3.37) with the dynamics of the relative position subsystem (3.25), we get the dynamics of the complete visual system:

$$\dot{\xi} = J_s\, u\,, \qquad J_s = \begin{bmatrix} S_z\, J_z \\ S_r\, J_r \end{bmatrix}\,, \tag{3.38}$$

where $J_s \in \mathbb{R}^{3\times n_m}$ is the system Jacobian, in this case.

Let $\tilde{\xi} := \xi_d(t) - \xi$ be the error state vector, expressed in terms of the generalized coordinates $\tilde{p}_{v_{\mathcal{H}}}$ and the relative position errors as follows:

$$\tilde{\xi} = \begin{bmatrix} \xi_{zd} \\ \xi_{rd} \end{bmatrix} - \begin{bmatrix} \xi_z \\ \xi_r \end{bmatrix}\,, \tag{3.39}$$

From (3.34), (3.37) and (3.39), the dynamics of the error state vector is governed by:

$$\dot{\tilde{\xi}} = \begin{bmatrix} \dot{\xi}_{zd} \\ \dot{\xi}_{rd} \end{bmatrix} - J_s\, u\,. \tag{3.40}$$

Now, we proceed to the derivation of a suitable control law in the joint space.

Theorem 3.3.4 *Consider the visual servoing system (3.38) with initial relative finger state 3.24 $\xi_r(0)$ and full-rank system Jacobian 3.38. Also, consider that:*

- *the object is firmly grasped by the robot hand at time $t = 0$;*
- *kinematic control assumption 3.15 is valid;*
- *relative state reference is constant ($\dot{\xi}_{rd} = 0$);*
- *the camera calibration parameters are fully known.*

The joint control signal that ensures the asymptotic convergence of the error state vector 3.39 to zero is:

$$u = J_s^{\dagger} \begin{bmatrix} \dot{\xi}_{zd} + \Lambda_z\, \tilde{\xi}_z \\ \Lambda_r\, \tilde{\xi}_r \end{bmatrix}\,, \tag{3.41}$$

where $\Lambda_z = \Lambda_z^T$ and $\Lambda_r = \Lambda_r^T$ are positive-definite gain matrices and $J_s^{\dagger}$ is the Moore-Penrose pseudo-inverse of the system Jacobian.

Proof 5 *See Appendix A - A.5.*

3.3.5 Image-based visual servoing control

Here, we present the control design for the image-based visual servoing control scheme, or IBVS, as well as for the finger relative position control scheme, as secondary objective.

The control error is defined only in terms of image space variables, and also in terms of the fingers relative positions.

Let a_{v_0} be the image projected area of the target object at time $t_0 < t$, such that $a_{v_0} = a_v(t_0)$, which corresponds to the uncertain depth $z_{c_0} = z_c(t_0)$. Back to section 3.1.2, we can solve (3.7) for the estimated object depth z_c as a function of the projected area at any time $t > 0$:

$$z_c(a_v) = z_{c0} \left(\frac{a_{v0}}{a_v} \right)^{\frac{1}{2}} .$$

From section 3.1.2 and 3.3.2, define the system state vector as

$$\xi := \begin{bmatrix} \xi_v \\ \xi_r \end{bmatrix} , \tag{3.42}$$

where $\xi_v = S_v \, [\, x_v \;\; y_v \;\; a_v \,]^\mathsf{T}$ is the image-based state vector, $\xi_r = S_r \, p_r$ is the relative position state vector, as usual, and $S_z \in \mathbb{R}^{m_v \times 3}$, $S_r \in \mathbb{R}^{m_r \times 6}$, $m_v, m_r \leq 3$ are appropriate selection matrices.

Taking the time-derivative of ξ_v and using (3.1), (3.6), and (3.13), we obtain:

$$\dot{\xi}_v = S_v \, \Omega_v \, \Big[R_{cp} \, J_{Po} \, \dot{\theta}_a + \mathcal{Q}(\omega_{cp}) \, R_{cp} \, p_o + \dot{r}_{cp} \Big] , \tag{3.43}$$

with

$$\Omega_v(p_v, a_v) = \frac{1}{z_c(a_v)} \begin{bmatrix} f \, \alpha_x & 0 & -x_v + x_{v_0} \\ 0 & f \, \alpha_y & -y_v + y_{v_0} \\ 0 & 0 & -2 \, a_v \end{bmatrix} , \tag{3.44}$$

where $z_c(a_v)$ is evaluated by means (3.42). The matrix $\Omega_v \in \mathbb{R}^{3 \times 3}$ is the well-known IBVS interaction matrix.

Once again, the subsystem described by the (3.43) is quite general, accounting for the relative motions of the camera frame $\mathcal{F}_c$ with respect to the palm frame $\mathcal{F}_p$. Next, considering the kinematic control approach (3.15), and once again supposing that there is no relative motion between the camera and the robot hand, and thus

$\dot{r}_{cp} = 0$ and $\omega_{cp} = 0$, yields:

$$\dot{\xi}_v = S_v\, J_v\, u\,, \qquad J_v(p_v, a_v, \theta_a) = \Omega_v\, R_{cp}\, J_{Po}\,. \tag{3.45}$$

where $J_v \in \mathbb{R}^{3\times n_m}$ is the so-called IBVS image Jacobian.

Combining (3.45) with the dynamics of the relative position subsystem (3.25), we get the dynamics of the complete visual system:

$$\dot{\xi} = J_s\, u\,, \qquad J_s = \begin{bmatrix} S_v\, J_v \\ S_r\, J_r \end{bmatrix}\,. \tag{3.46}$$

where $J_s \in \mathbb{R}^{(m_v+m_r)\times n_m}$ is the system Jacobian, in this case.

Let $\tilde{\xi} := \xi_d(t) - \xi$ be the error state vector, expressed in terms of the image space errors $\tilde{\xi}_v$ and the relative position errors as follows:

$$\tilde{\xi} = \begin{bmatrix} \xi_{vd} \\ \xi_{rd} \end{bmatrix} - \begin{bmatrix} \xi_v \\ \xi_r \end{bmatrix}\,. \tag{3.47}$$

From (3.42), (3.45) and (3.47), the dynamics of the error state vector is governed by:

$$\dot{\tilde{\xi}} = \begin{bmatrix} \dot{\xi}_{vd} \\ \dot{\xi}_{rd} \end{bmatrix} - J_s\, u\,. \tag{3.48}$$

Now, we proceed to the derivation of a suitable control law in the joint space.

Theorem 3.3.5 *Consider the visual servoing system given by (3.46) with relative finger state 3.24 $\xi_r(0)$, and full-rank system Jacobian (3.46). Also, consider that:*

- *the object is firmly grasped by the robot hand at time $t = 0$;*
- *kinematic control assumption 3.15 is valid;*
- *relative state reference is constant ($\dot{\xi}_{rd} = 0$);*
- *the camera calibration parameters are fully known.*

The joint control signal that ensures the asymptotic convergence of the error state vector 3.47 to zero is:

$$u = J_s^{\dagger} \begin{bmatrix} \dot{\xi}_{vd} + \Lambda_v\, \tilde{\xi}_v \\ \Lambda_r\, \tilde{\xi}_r \end{bmatrix}\,, \tag{3.49}$$

where $\Lambda_v = \Lambda_v^T$ and $\Lambda_r = \Lambda_r^T$ are positive-definite gain matrices and $J_s^{\dagger}$ is the Moore-Penrose pseudo-inverse of the system Jacobian.

Proof 6 *See Appendix A - A.6.*

3.4 Singularity Analysis

In this section, we study the rank of the Jacobian matrices that appear in the systems dynamics presented in the previous sections. This study is going to be of particular importance for the performance of the control algorithms, since all of them are based on the kinematic control approach and the linearization of the error dynamics through multiplication by the Moore-Penrose right pseudo-inverse of the corresponding system Jacobian.

We must be sure that all Jacobian matrices are well conditioned, avoiding the occurrences of matrix *singularities* that can create computational problems during the inversion of these matrices.

3.4.1 Hand Jacobian singularities

From the expression of the hand Jacobian for the Robotiq hand in (2.53), we can deduce some important situations where its rank is reduced.

First, note that it is naturally rank deficient, since all elements in the first line are null. This fact is due to the structure of the first finger, which cannot perform movements in the x-direction of the palm frame $\mathcal{F}_p$. This fact is important, because it limits the number of possible choices for the selection matrices used to restrict the directions for control. We should choose only valid directions when applying the control algorithms, otherwise the system Jacobian matrix will be rank deficient and therefore, the stability of the closed-loop system is going to be compromised.

Then, note that when $\delta_{k1} = 0$ and $\delta_{k2} = 1$ for $k = 1, 2, 3$, all elements in several lines of J_h become null, meaning that the instantaneous change in position or orientation on the corresponding direction is stationary in that particular configuration. In other words, the fingers lose *manipulability* in this configuration.

Another issue is about the configuration of the finger angles θ_k, $k = 1, 2, 3$ as defined in (2.40). If θ_{kj}, $k = j = 1, 2, 3$ are all zero, all sines in the third line of the complete finger Jacobian matrices J_F^k are null. This line is related to the z-component of the fingers velocities, and this is a singular configuration because when the fingers are all stretched (corresponding to null θ_{kj}), they cannot instantaneously move in the z-direction. By observing the types of grasping in Fig. 2.3, we note that the fingers must be at least *slightly stretched* to perform a manipulable grasp, even for small objects. Because of this fact, we can expect to reach singular configurations of the kind described above if we select the z-direction for control for any manipulation task. In fact, even for independent control of the fingers, selecting the z-direction for control means that there will be singular configurations in the operational space.

To overcome these issues, we can select the y-direction, for both fingers and object position control. In fact, if $\theta_{ka_{min}} \geq \theta_{ka} \leq \gamma_2$ for $k = 1, 2, 3$ during any task, the second line elements of the J_F^k are always non-null ($\theta_{ka_{min}}$ is the minimum value for θ_{ka}).

Under these limitations for θ_{ka}, we could also select the fourth line of J_F^k for control, which is the equivalent of controlling the *roll* angle of the k-th fingertip frame $\mathcal{F}_f$.

With respect to fingers 2 and 3, we can select the *pitch* angle for shared control, corresponding on the fifth line of J_F^2 and J_F^3, which are non-null for all finger configuration.

3.4.2 Grasp matrix singularities

In this section, we study the rank of the *grasp matrix* G_o, defined in (2.29). The grasp matrix is defined as:

$$G_o = A^{+}(I - H^{\mathsf{T}}\,(\tilde{A}\,H^{\mathsf{T}})^{+}\,\tilde{A})\,.$$

To guarantee that G_o is full rank, it is enough to guarantee that H^{T}, A and $\tilde{A}$ are full rank. For the Robotiq gripper, the Adjoint matrix A is constructed from the stacking of the three Adjoint matrices A_1, A_2 and A_3, which are supposed to be full rank:

$$A = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix},$$

These matrices transforms vectors written in the contact frame $\mathcal{F}_{ck}$ to the corresponding vectors in the object frame $\mathcal{F}_o$. Let $\tilde{A}$ be the annihilator matrix of A. Defining

$$\tilde{A} = \begin{bmatrix} \alpha\,A_1^{-1} & \beta\,A_2^{-1} & \gamma\,A_3^{-1} \end{bmatrix}, \tag{3.50}$$

we can left multiply it by A, yielding:

$$\tilde{A}\,A = (\alpha + \beta + \gamma)\,I\,.$$

Since, by definition, $\tilde{A}\,A = 0$, an annihilator matrix defined by (3.50) is non-singular for any combination of α, β, γ in which $\alpha + \beta + \gamma = 0$. In this work, we have chosen $\alpha = 1$, $\beta = 0.5$ and $\gamma = -0.5$.

The expressions for the contact matrices $H_k \in \mathbb{R}^{6\times n_r}$, $n_r < 6$ are obtained by the vertical stacking of $\mathbb{R}^{6-n_r\times n_r}$ zero matrices and identity matrices of order n_r. Thus, all H_k are always of full column rank.

3.4.3 Stacked Jacobian singularities

Most of the control designs developed in the previous chapter make use of a strategy consisting in building a combined state vector from two unrelated systems, such as vision and prehension. Here, we cover an important result regarding the rank analysis of the stacked matrices that appear in the dynamics of such combined systems.

Proposition 2 *A matrix J, constructed from the vertical stacking of J_1 and J_2, as:*

$$J = \begin{bmatrix} J_1 \\ J_2 \end{bmatrix}$$

is full rank if and only if the matrices J_1, J_2, $J_1\,P_{J_2}^{\perp}\,J_1^{\mathsf{T}}$ and $J_2\,P_{J_1}^{\perp}\,J_2^{\mathsf{T}}$ are all full rank, where $P_{J_1}^{\perp}$ and $P_{J_2}^{\perp}$ are the projection matrices of J_1 and J_2, defined by:

$$\begin{aligned} P_{J_1}^{\perp} &= I - J_1^{\dagger}\,J_1\,, \\ P_{J_2}^{\perp} &= I - J_2^{\dagger}\,J_2\,. \end{aligned}$$

The † operator denotes the right pseudo-inverse matrix operator.

Proof 7 *See Appendix A - A.7.*

Chapter 4

Experimental and Simulation Results

In this chapter, we present experimental and simulation results in order to illustrate the performance and effectiveness of the proposed methodology in terms of the modeling and control design. We also present some experimental data for the validation of the proposed empirical model for the fingers kinematic restrictions. First, we present the experimental setup, as long as the algorithms and programming environment utilized for the experiments and for the simulations. In the next sections, the results of five different experiments performed with the Robotiq hand are illustrated and discussed. The results of a simulation experiment are also shown. The first experiment consists in the validation of the empirical relations (2.43), (2.43), (2.43) and (2.44), proposed in section 2.6.2. The next five experiments are based on the control approaches for a robot hand developed in section 3.3.1, applied to the Robotiq hand. The last experiment consists in a control simulation of a 12 Dof robot hand with similar structure than the Robotiq hand. Finally, we present some interpretations about the obtained results, discussions about the feasibility of the presented theoretical framework and also about the performance of the developed control schemes.

4.1 Experimental Setup

The 3-Finger Adaptive Robot Gripper is a programmable and mechanically intelligent robot hand, which is able to manipulate a wide variety of objects. The Robotiq hand has easy integration, being compatible with all major industrial robot manufacturers, such as ABB and Yaskawa. Mechanical and software components are provided to make the installation and use of the device easier. In addition, a number of ROS packages are also available for download. The 3-finger model is rugged and agile, designed to operate in unstructured environments. By means of the Robotiq Graphical User Interface it is possible to control force, speed and position parameters for each finger, thanks to many options of communication protocols. Among several applications, the Robotiq hand can be used for advanced manufacturing and research as well as machine tending. The main mechanical and electrical specifications are available in Tab. 4.1. A complete documentation covering all specifications of the gripper can be found in (ROBOTIQ, 2013).

Gripper opening, mm	0 to 155
Maximum recommended payload, kg	2.5
Grip force, N	15 to 60
Closing speed, mm/s	22 to 110
Finger position repeatability, mm	0.05
Nominal supply voltage, V	24
Maximum total current, A	1.5

Table 4.1: Specifications of the 3-Finger Adaptive Gripper.

All experiments were performed on a desktop PC with Intel Core i5 Processor (8M Cache, 2.8 GHz) 4GB RAM, running Windows OS. The control algorithms were implemented in Matlab (The MathWorks Inc.) Release 2013a and a proper class was developed, implementing a set of methods designed to execute the control loop, send/receive data to/from the Robotiq gripper and perform all necessary calculations. An USB connection was created using the Matlab Serial Port class, and the communication link with the Robotiq gripper was established by the Modbus RTU communication protocol. These codes are open source and can be found in https://github.com/teteuZ/GripperKinematicControl and https://github.com/teteuZ/ServoPrehensionControl.

The transferred/received data include bytes representing angular position, angular velocity and current in the motors, as long as fault bit flags for each finger. The data conversion between the angular position and velocity of the motor shafts, θ_a (in radians) and $\dot{\theta}_a$ (in radians per second), and the corresponding sent/received one-byte values $a_j, b_j \in [0, 255]$ are given empirically by ($k = j = 1, 2, 3$):

$$\theta_{ka} \;=\; \begin{cases} \;\;0.0071\, a_k - 0.0419\,, & k\!=\!1,2,3\,, \\ -0.0023\, a_k - 0.3112\,, & \quad k\!=\!4\,, \end{cases}$$

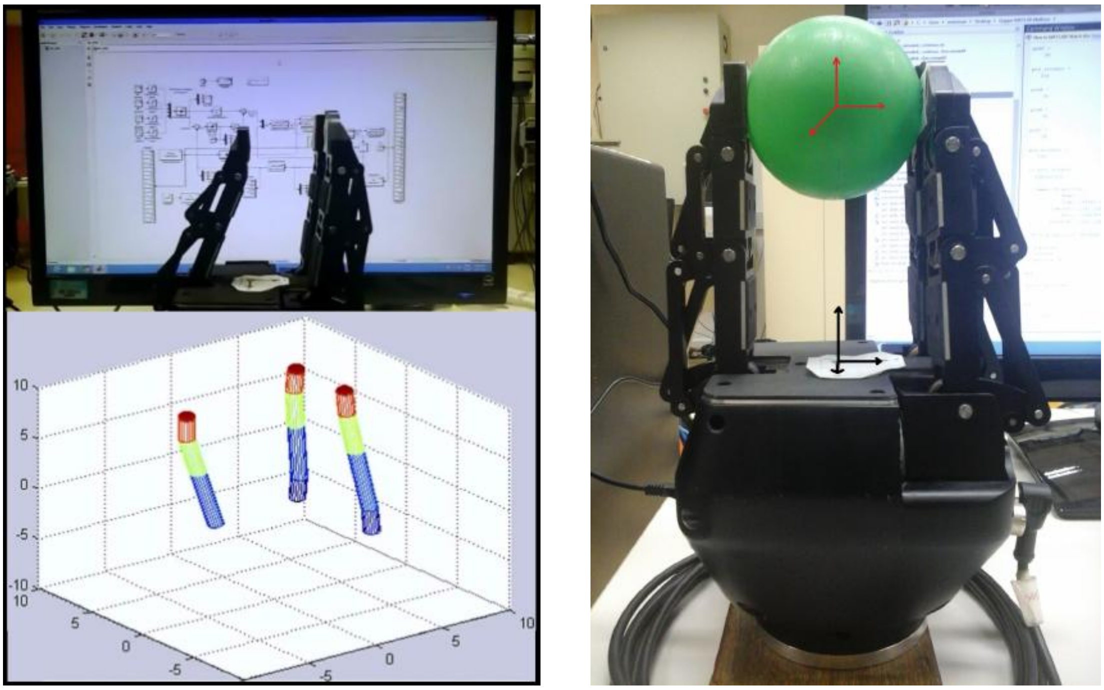
$$|\dot{\theta}_{ka}| \;=\; \begin{cases} 0.0290\, b_k + 0.1884\,, & k\!=\!1,2,3\,, \\ 0.0100\, b_k + 0.0650\,, & \quad k\!=\!4\,. \end{cases}$$

The object manipulation task was performed using plastic balls with a radius varying from $r = 30\ mm$ to $r = 36\ mm$. In each experiment, the Robotiq gripper was set up in the "pinch" mode (Fig. 2.3) and an *ad-hoc* Matlab routine closes the fingers around the object until it is firmly grasped.

With respect to the prehension control, from the forward kinematics map, a routine calculates the initial relative position between fingers 1 and 2 and between fingers 2 and 3. Because of the limited number of DoFs in the Robotiq hand, the relative positions p_{12} and p_{23} are only controlled in the y-axis and x,y-axis of the palm frame, respectively. This way, the corresponding selection matrices for the relative states are:

$$S_{12} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, \qquad S_{23} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \tag{4.1}$$

Before being tested on the real robotic system, the performance of the control algorithms was verified in a 3D graphic animation, also developed in Matlab. In the experiments, a sampling time of approximately $h = 0.1\ s$ ensures a satisfactory communication between the software and the gripper. However, this communication rate can affect the stability of the algorithms under high eigenvalues of the gain matrices. Fig. 4.1a illustrates the experimental setup (on top) and the simulation environment (on bottom) used to carry out the independent motion task. Fig. 4.1b shows the initial configuration of the robot hand and the ball used to perform the manipulation task.



(a) Exp. setup for independent finger control. (b) Exp. setup for object manipulation tasks.

Figure 4.1: Experimental setup for kinematic experiments.

The experimental setup arranged for the visual servoing experiments is shown in Fig. 4.2. Here, we use a set strongly colored balls, because the computer vision algorithm used to identify the ball in the scene is based in color extraction.

Some of the main Microsoft Kinect hardware specifications can be found in Tab. 4.2, and a complete documentation covering all specifications of the Kinect sensor can be found in (M.R.ANDERSEN *et al.*, 2012).

In Fig. 4.3, we can observe the graphical interface used for the simulation of a 12 Dof robot hand, whose structure is similar to the Robotiq hand, if every joint angle could be directly controlled by a proper motor (no kinematic constraints in the fingers).

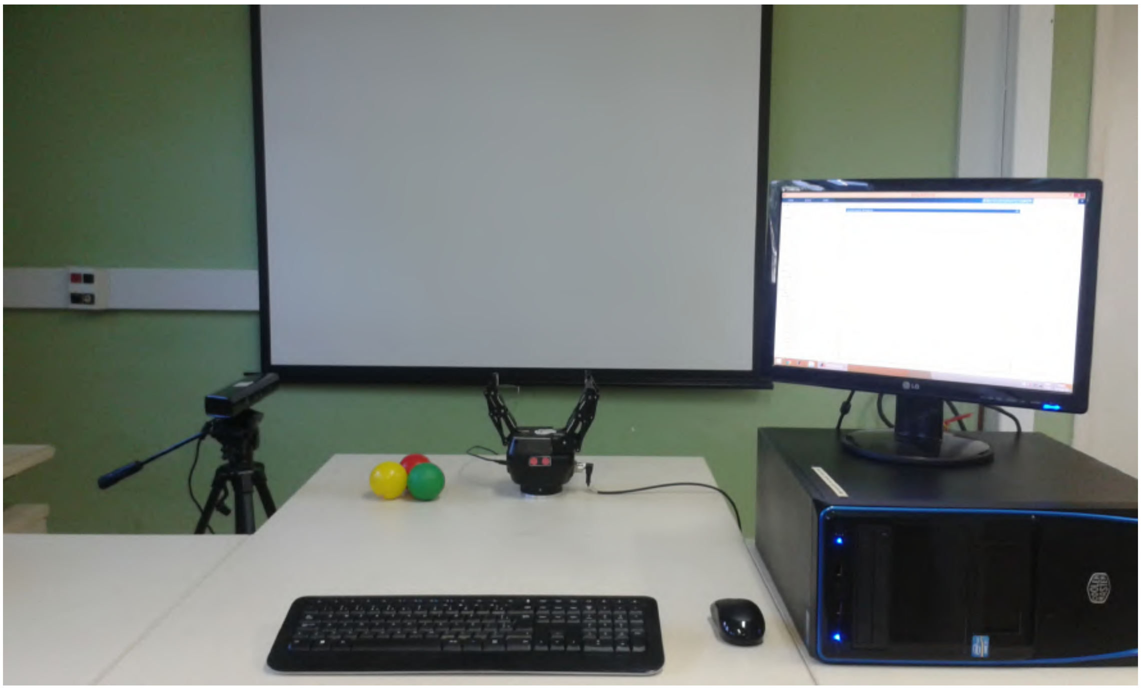


Figure 4.2: Experimental setup for visual servoing experiments.

Property	Value
Angular Field-of-View	57° horz., 43° vert.
Vertical tilt range	±27°
Framerate	approx. 30Hz
Nominal spatial range	640 × 480 (VGA)
Nominal spatial resolution (at 2m distance)	3mm
Nominal depth range	0.8 – 3.5m
Nominal depth resolution (at 2m distance)	1cm
Device communication type	UBS (+ external power)

Table 4.2: Microsoft Kinect hardware specifications.

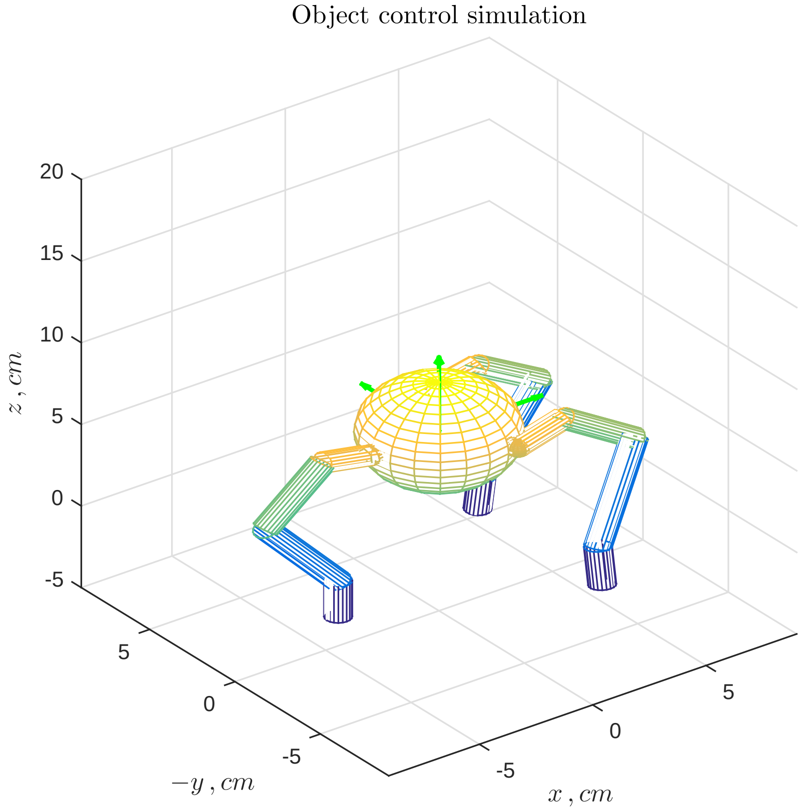


Figure 4.3: Simulation of a 12 Dof Robot Hand.

4.2 Experimental Results

In this section, we present all the experimental results obtained with the 3-Finger Adaptive Robotiq Gripper. First, we present experimental data for validation of the finger kinematic constraints introduced in section 2.6.2. Then, we proceed with four control experiments: independent finger control, object position control and visual servoing control experiments, divided into *hybrid visual servoing* (HVS) and *image-based visual servoing* (IBVS) methodologies. The last section presents the simulation results for a 12 Dof robot hand, a simulated version of the Robotiq gripper with no finger kinematic constraints and an extra joint.

Table 4.3 summarizes all results of experiments and simulations presented in this chapter.

Experiment/Simulation	**Section**	**Results**
Validation of the Empirical Model	Sec. 4.2.1	Fig. 4.5
Finger Control Experiments	Sec. 4.2.2	Fig. 4.6
Object Control Experiments	Sec. 4.2.3	Fig. 4.7
HVS Control Experiments	Sec. 4.2.4	Figs. 4.10, 4.11
IBVS Control Experiments	Sec. 4.2.5	Figs. 4.12, 4.13
Simulation of a 12 Dof Robot Hand	Sec. 4.2.6	Figs. 4.14, 4.15

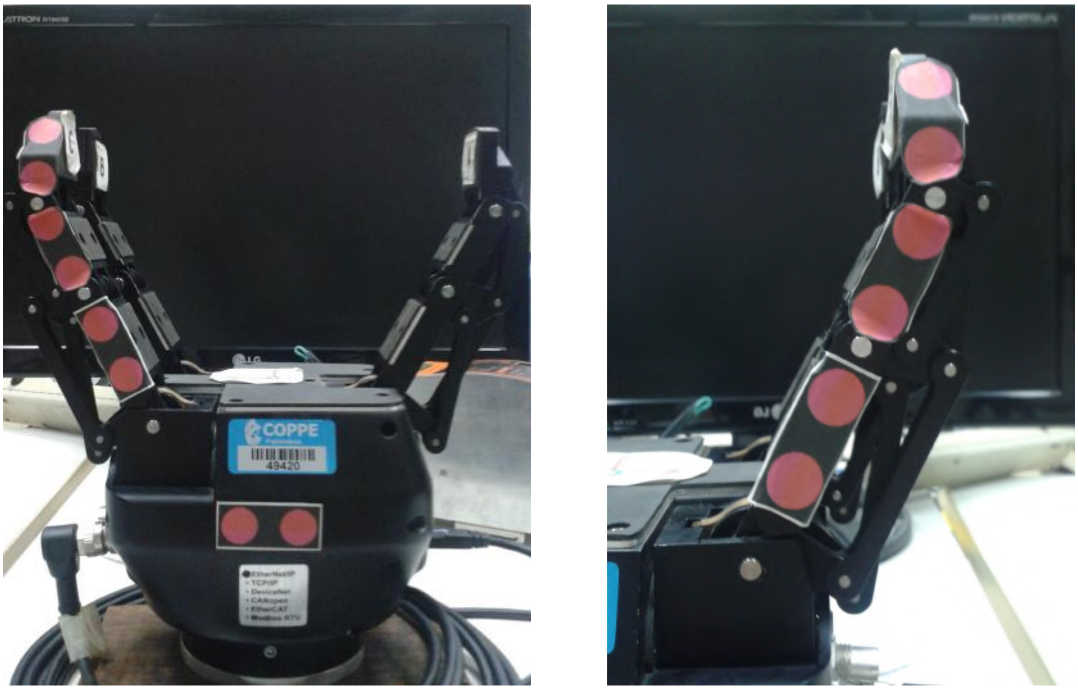
Table 4.3: Control experiments.

4.2.1 Validation of the Empirical Model for the Finger Joints

In this section, we propose a validation experiment, where we want to determine the set of model parameters α_k, β_k, γ_k, $k = 1, 2, 3$ proposed in section 2.6.2 by means of a visual experiment and compare it with the results obtained empirically.

The experiment consists in tracking four pairs of colored targets attached to the robot hand with a video camera: one pair is attached in one side of the palm and the other three are attached in each phalanx of a chosen finger.

This setup can be visualized in Fig. 4.4. We can see that the pairs of targets are aligned with the corresponding phalanx link, and the pair on the palm is positioned horizontally. The robot hand is set in the "Basic mode", and the video camera is positioned in way that the image plane is parallel to the y-z plane of the palm frame. A color based algorithm continuously extracts the red components of the camera stream, finding the targets in the image. The algorithm calculates the angles between the lines formed the pairs of points (x_{vi}, y_{vi}) in each phalanx and the line formed by the pair on the palm. By slowly opening and closing the chosen finger, we can obtain a series of angle measurements. This way, we the can visually determine the validity of the proposed empirical model and its parameters, by comparing the functions of the empirical model with the measured behavior of how the angles of



(a) Pairs of tags in the fingers / palm.

(b) Finger tags in detail.

Figure 4.4: Colored tags.

the phalanges change when the corresponding motor angle changes.

The empirical relations presented in section 2.6.2 are repeated here for convenience:

$$\theta_{k1} = \begin{cases} \alpha_1\,\theta_{ka} + \beta_1\,, & \theta_{ka} \leq \gamma_1\,, \\ \eta_1\,, & \theta_{ka} > \gamma_1\,, \end{cases}$$

$$\theta_{k2} = \begin{cases} \eta_2\,, & \theta_{ka} \leq \gamma_2\,, \\ \alpha_2\,\theta_{ka} + \beta_2\,, & \theta_{ka} > \gamma_2\,, \end{cases}$$

$$\theta_{k3} = \begin{cases} \alpha_3\,\theta_{k1} + \beta_3\,, & \theta_{ka} \leq \gamma_3\,, \\ \eta_3\,, & \theta_{ka} > \gamma_3\,, \end{cases}$$

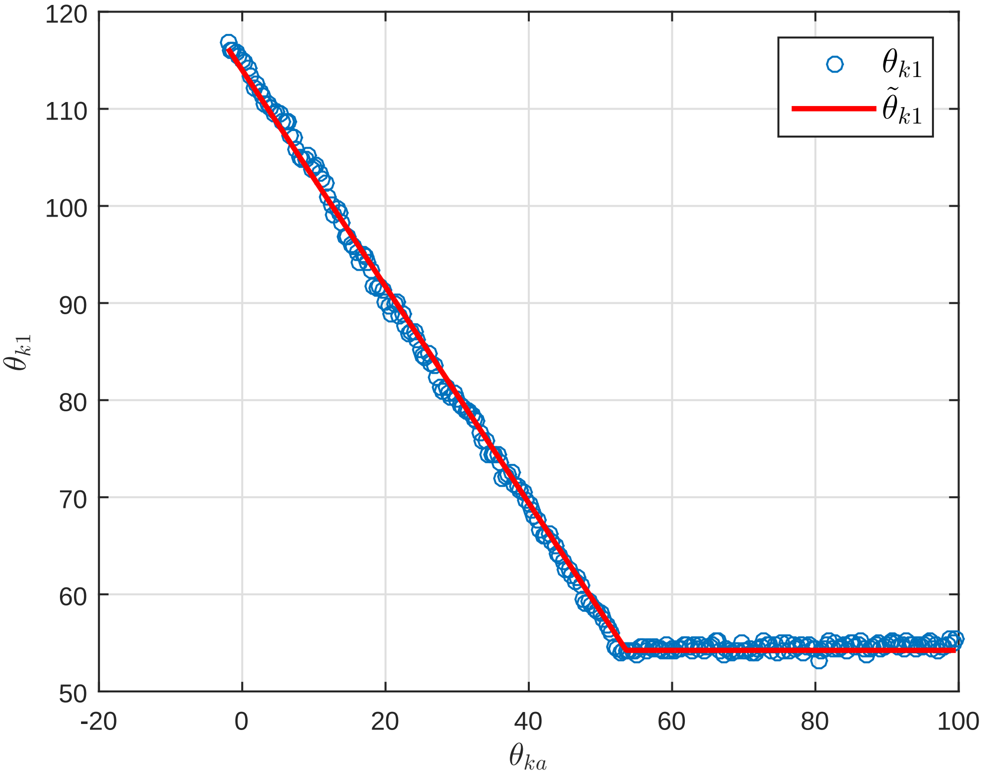
A reasonable condition that these functions must met to represent the behavior of a real mechanical system is to be at least of class C^0 (continuous). This restriction allows us to calculate the η_k parameters, imposing the continuity condition to θ_{kj},

$j = 1, 2, 3$, yielding:

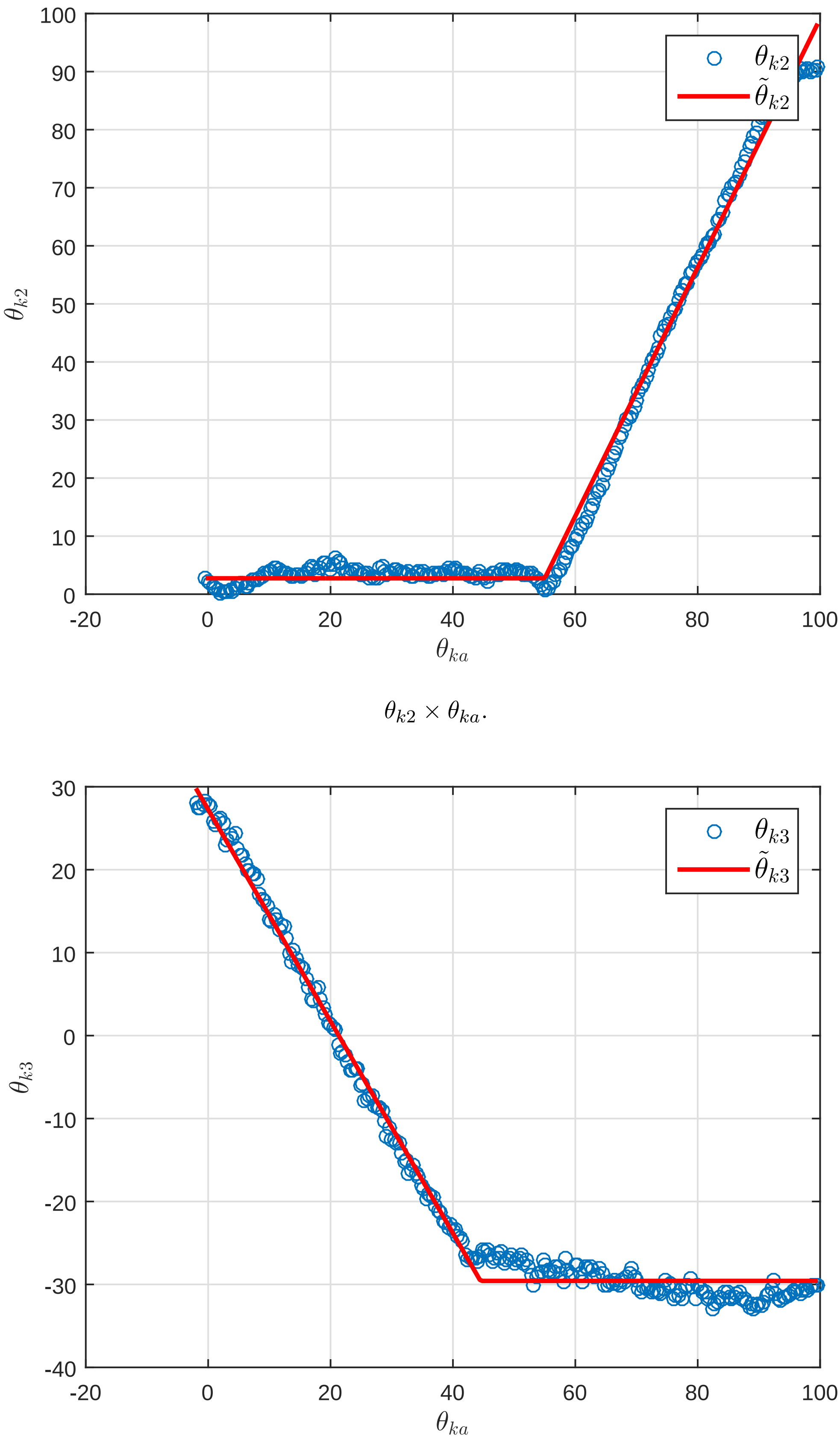
$$\begin{aligned} \eta_1 &= \alpha_1 \gamma_1 + \beta_1 , \\ \eta_2 &= \alpha_2 \gamma_2 + \beta_2 , \\ \eta_3 &= \alpha_3 (\alpha_1 \gamma_3 + \beta_1) + \beta_3 , \end{aligned} \tag{4.2}$$

where η_3 has this form because it is known from observation that $\gamma_3 < \gamma_1$. In particular, it is also known that $\eta_2 \approx 0$, $\alpha_3 \approx 1$, and $\beta_3 \approx 0$. Since the γ_k represents the angular limits of θ_{ka} in which each phalanx angle suddenly changes its behavior, we expect to observe these discontinuity points in the angular data.

A least squares algorithm is utilized to estimate α_k and β_k from the data points, and equation (4.2) is used to calculate the η_k constants, provided that the discontinuity points η_k are known. Fig. 4.5 shows the results of the experiments. The red lines are the predicted empirical functions and the blue dots represent the measured data. The results are plotted in degrees, for simplicity.



$\theta_{k1} \times \theta_{ka}$.



$\theta_{k2} \times \theta_{ka}$.

$\theta_{k3} \times \theta_{ka}$.

Figure 4.5: Results of the validation experiments.

First, only the measured pairs of angles were plot, allowing us to determine the set of discontinuity points γ_k (in radians):

$$\gamma_1 \;=\; 0.935\,, \qquad \gamma_2 \;=\; 0.959\,, \qquad \gamma_3 \;=\; 0.776\,. \tag{4.3}$$

Then, the least squares algorithm was used to determine the following set of parameters, used to plot the functions in red in Fig. 4.5 (β_k and η_k are in radians):

$$\begin{aligned} \alpha_1 &= 1.1149\,, & \beta_1 &= -0.4187\,, & \eta_1 &= 0.6243\,, \\ \alpha_2 &= 2.1417\,, & \beta_2 &= -2.0082\,, & \eta_2 &= 0.0477\,, \\ \alpha_3 &= -1.1454\,, & \beta_3 &= 0\,, & \eta_3 &= -0.5162\,. \end{aligned}$$

Although the results were very close to our expectations, there are several sources of errors that could lead to inaccurate results in this experiment; the camera alignment could be imperfect, non-modeled distortion effects caused by the lens curvature could change the real angles in the image, luminosity level changes could alter the color perception, leading to inaccurate tag center detection, among others.

4.2.2 Finger Control Experiments

In this section, we present the results for finger control experiments. Tracking experiments were performed, where we have selected the y-direction of fingers $1, 2$ and 3 and the x-direction of finger 2 for control. Thus, the selection matrices defined in section 3.3.1 are:

$$S_1 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, \qquad S_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \qquad S_3 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}.$$

and thus, the state vector is $\xi = [\,p_{1,y} \;\; p_{2,x} \;\; p_{2,y} \;\; p_{3,y}\,]^{\mathbf{T}}$.

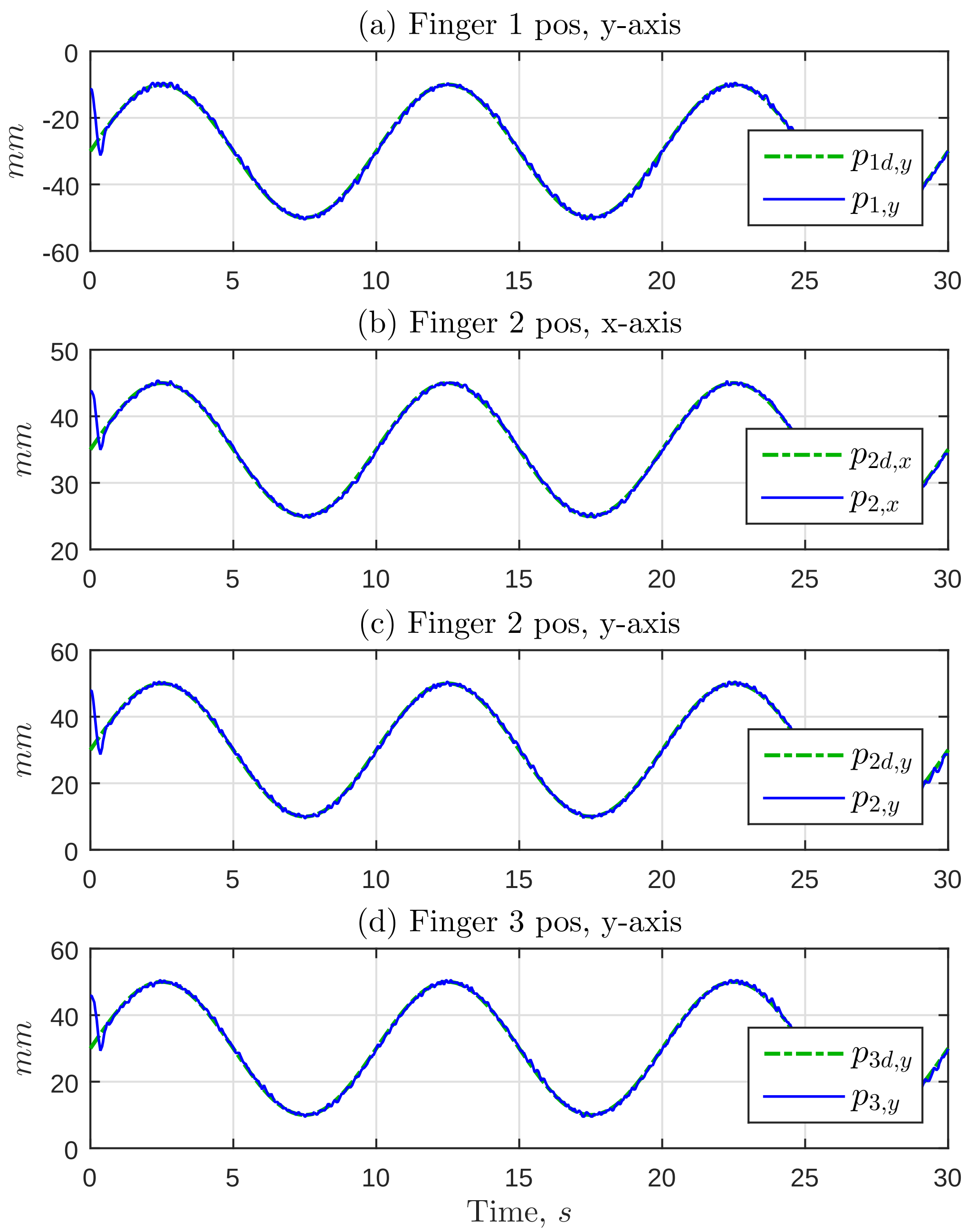
The gain matrix can be written as $\Lambda = \lambda\, I_4$, where $I_4 \in \mathbb{R}^{4\times 4}$ is the 4×4 identity matrix, and $\lambda = 80$ is a scalar gain. Our objective is to track a bounded, time-varying reference function. We have used the following sinusoidal reference signals:

$$\xi_{kd} = a_k\, sin(\omega\, t) + b_k\,, \qquad k = 1, 2, 3, 4\,.$$

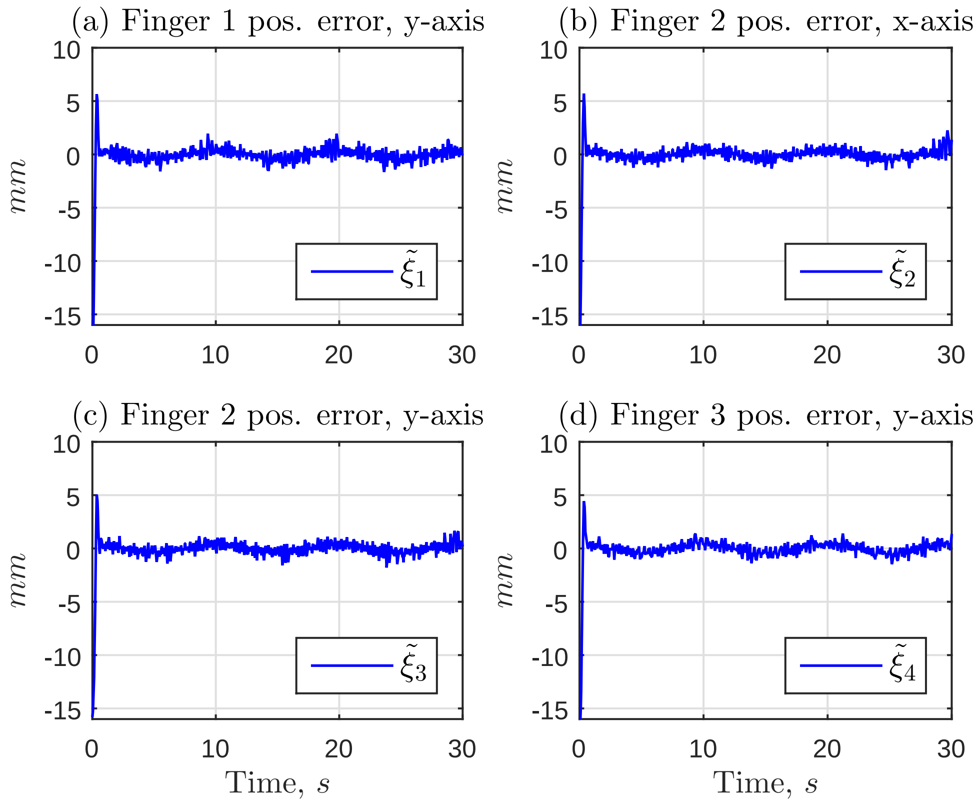
where $b_1 = -30\,, b_2 = 35\,, b_3 = 30\,, b_4 = 30\, mm$, $\omega = 0.4\,\pi\, rad\, s^{-1}$ and $a_k = 20\, mm$ for $k = 1, 2, 3, 4$, and $t \in [\,0,\, T_{max}\,]$ where $T_{max} = 30\, s$ is the total runtime of the experiment.

Sinusoidal functions provide a smooth, well-conditioned signal for testing control algorithms: being a time-variable, differentiable signal, their derivatives can be applied analytically on the velocity feed-forward term $\dot{\xi}_d$ in the control equation (3.21), avoiding computational problems with symbolic derivatives. Furthermore, they can also be used for testing the control systems bandwidth; by carefully adjusting the angular frequency of the reference sinusoid, we can determine the control system performance when subjected to fast input variations. The reference and position signals, error signals, position control signals and joint control signals are showed in Fig. 4.6.

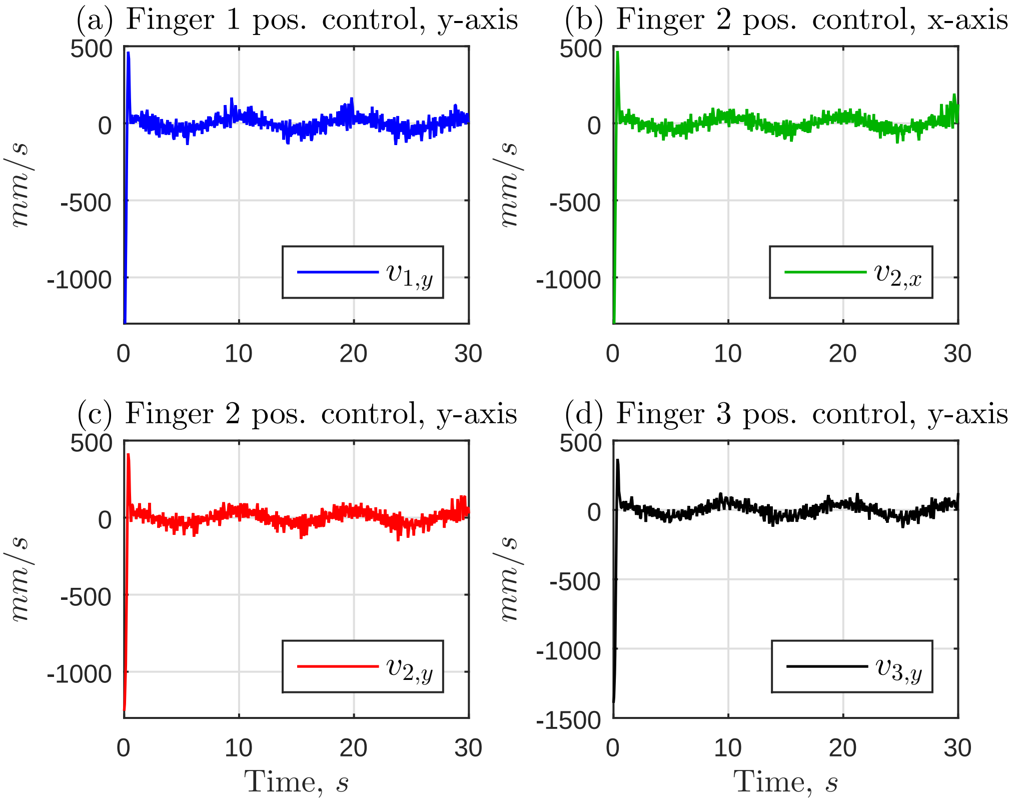
Note that the value chosen for the scalar gain λ provides a fast dynamic response to the system, ensuring the zero-convergence of the tracking error in less than $1\,s$, without affecting the convergence of the algorithm. The magnitude of the error signal is bounded to a small region of $3\,mm$, due to the response delay of the system, and except for the initial instant, the magnitude of the joint control signals also remains bounded in less than $3\,rad\,s^{-1}$, which is an acceptable value for the velocity of the motors, considering the angular frequency of the reference sinusoid.



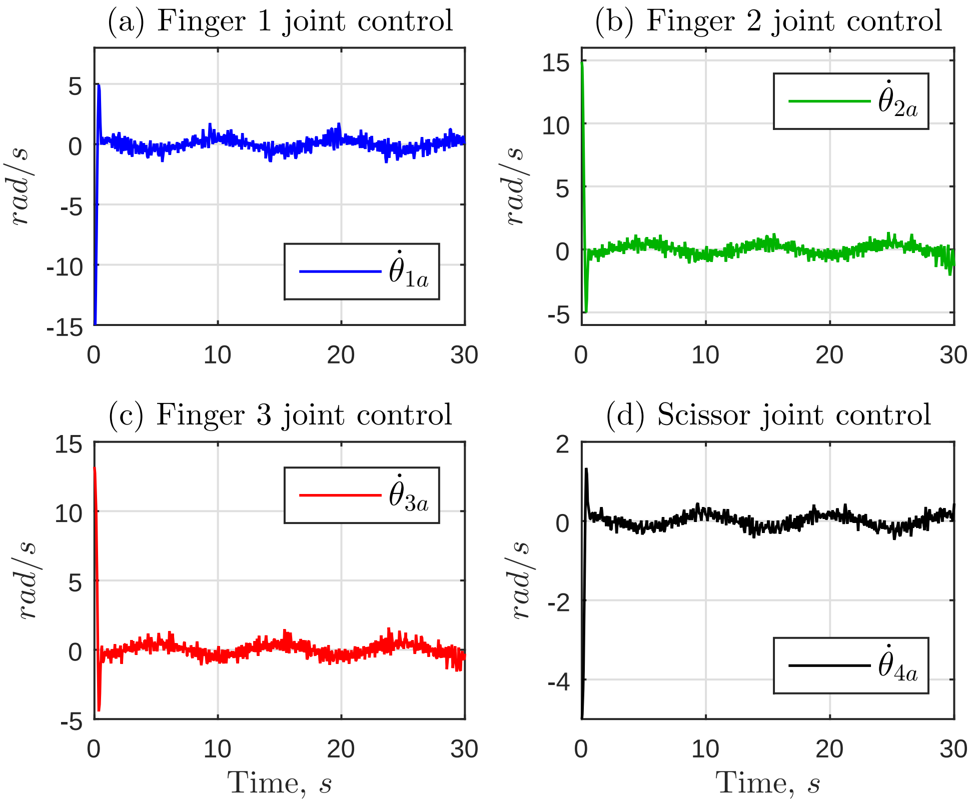
1. Reference and finger positions for the tracking experiment.



2. Error signals for the tracking experiment.



3. Position control signals for the tracking experiment.



4. Joint control signals for the tracking experiment.

Figure 4.6: Results for the tracking experiment.

4.2.3 Object Manipulation Experiments

The object grasping and manipulation task was performed using a plastic ball with a radius of $r = 36.2\ mm$. In this case, the Robotiq gripper is set up in the "pinch" mode and an *ad-hoc* Matlab routine closes the fingers around the object. Here, the grasping routine must successfully guarantee the initial object prehension. From the forward kinematics map, another routine calculates the initial relative position between the fingers 1 and 2, as well as between the fingers 2 and 3. Finally, another routine releases the object by opening the fingers, before the experiment is initiated. The object is held fixed on its initial position, so that the grasping parameters (i.e., the initial relative positions) do not change when the control algorithm starts running.

Since the plastic ball is a spherically symmetric object, and since orientation control is not the objective of this work, we made the following simplifying assumption about the rotation matrix R_{po}:

$$R_{po} \approx I_3 \, .$$

This assumption is valid in this particular manipulation task because, from Fig. 4.1b, for a small task space width, the ball's orientation remains approximately fixed.

The experiment consists in the tracking a sinusoidal trajectory in the y-axis direction of the Cartesian space. Thus the object selection matrix S_o is:

$$S_o = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} .$$

The selection matrices for the relative states are defined in (4.1), and the references for the relative states are constant values defined by the initial relative positions, obtained when the object was grasped. The object reference is defined by the analytical expression:

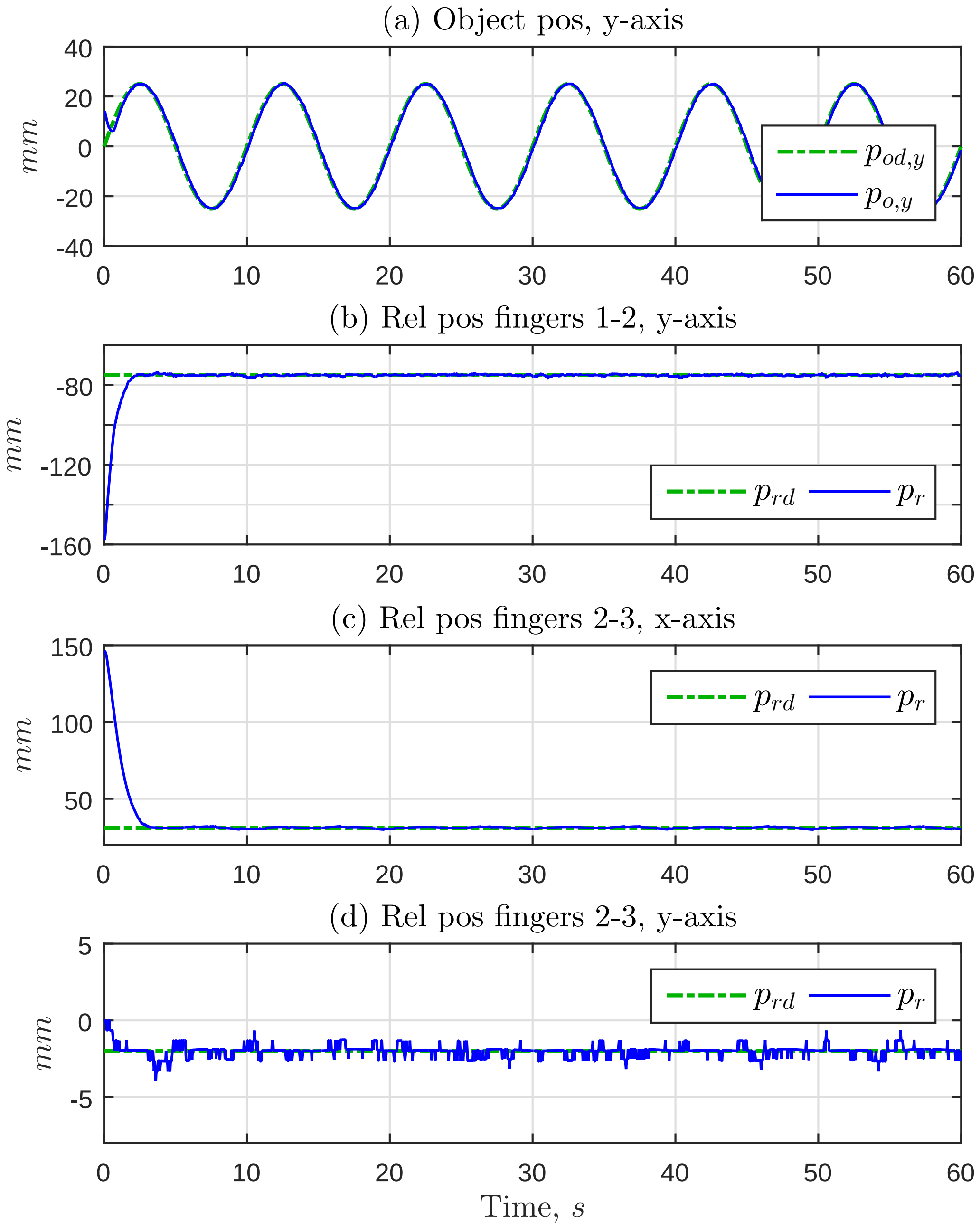
$$\xi_1 = S_o \, p_{od}(t) = a \, \sin(\omega \, t) + b \, ,$$

where a, ω and b are the amplitude, angular frequency and mean value of the sinusoidal signal, and $t \in [\, 0, \, T_{max} \,]$ where $T_{max} = 60 \, s$ is the total runtime of the experiment. The gain matrices are defined as $\Lambda_o = \lambda_o$ and $\Lambda_r = \lambda_r \, I_3$, and the desired relative position is given by:

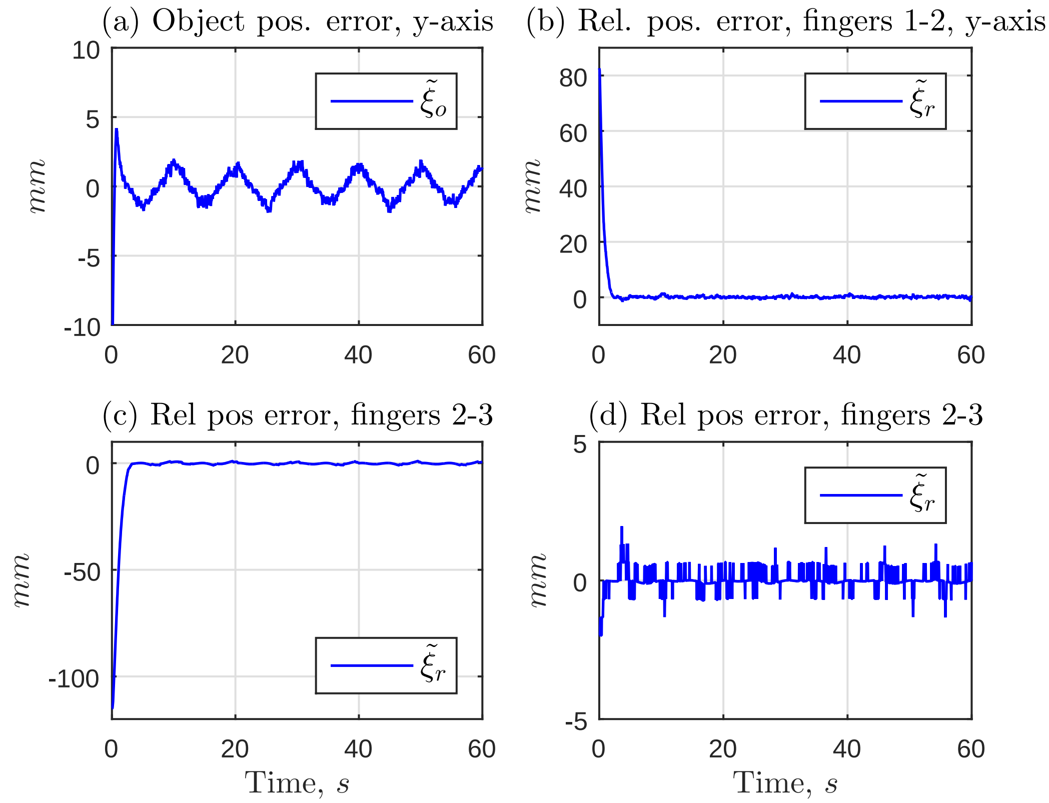
$$\xi_{rd} = S_r \, p_{rd} \approx [\,-75.1 \;\; 31.0 \;\; -1.9\,]^{\mathsf{T}} \; mm \, .$$

Initially, the fingers are set to full opening, and the object is held fixed on the same position until the experiment starts. This way, the Robotiq hand can regrasp the plastic ball on the beginning of the experiment, and the convergence of the relative state vector to its desired values can be observed.

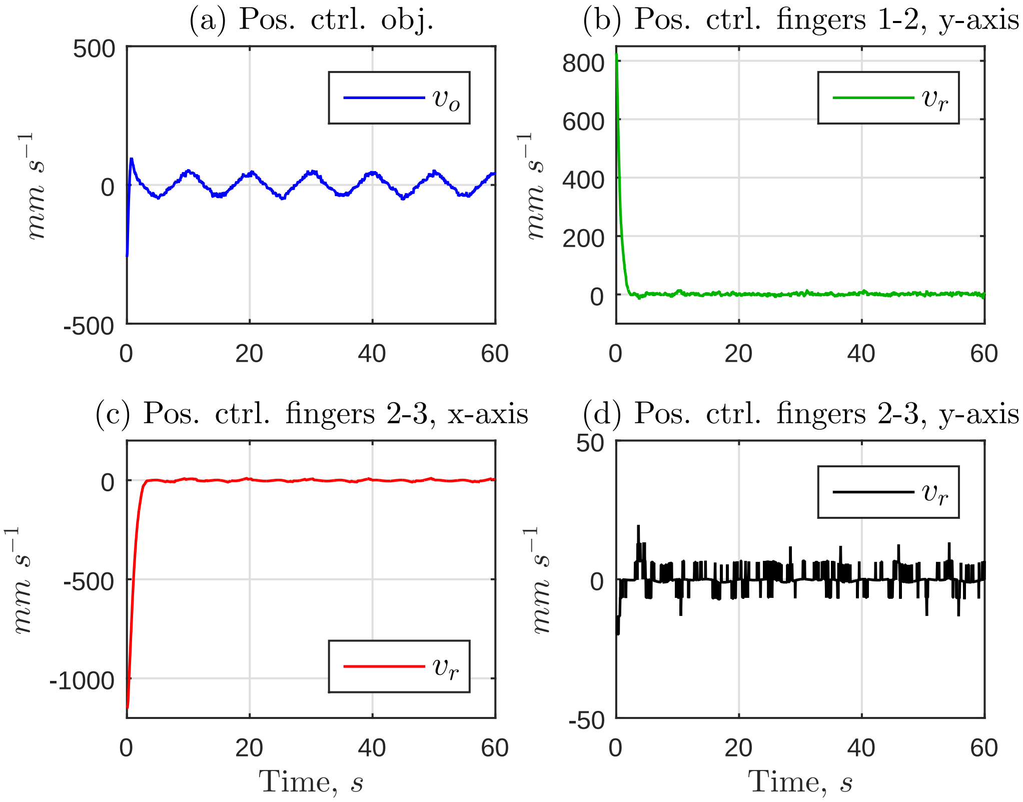
The scalar gains and the parameters of the sinusoid are set up to $\lambda_o = 20 \, s^{-1}$, $\lambda_r = 10 \, s^{-1}$, $a = 25 \, mm$, $\omega = \frac{\pi}{5} \, rad \, s^{-1}$ and $b = 0$. The results are shown in Fig. 4.7, where it is possible to observe the time history of the trajectory tracking, the relative position regulation, the position errors, and the control signals, respectively.



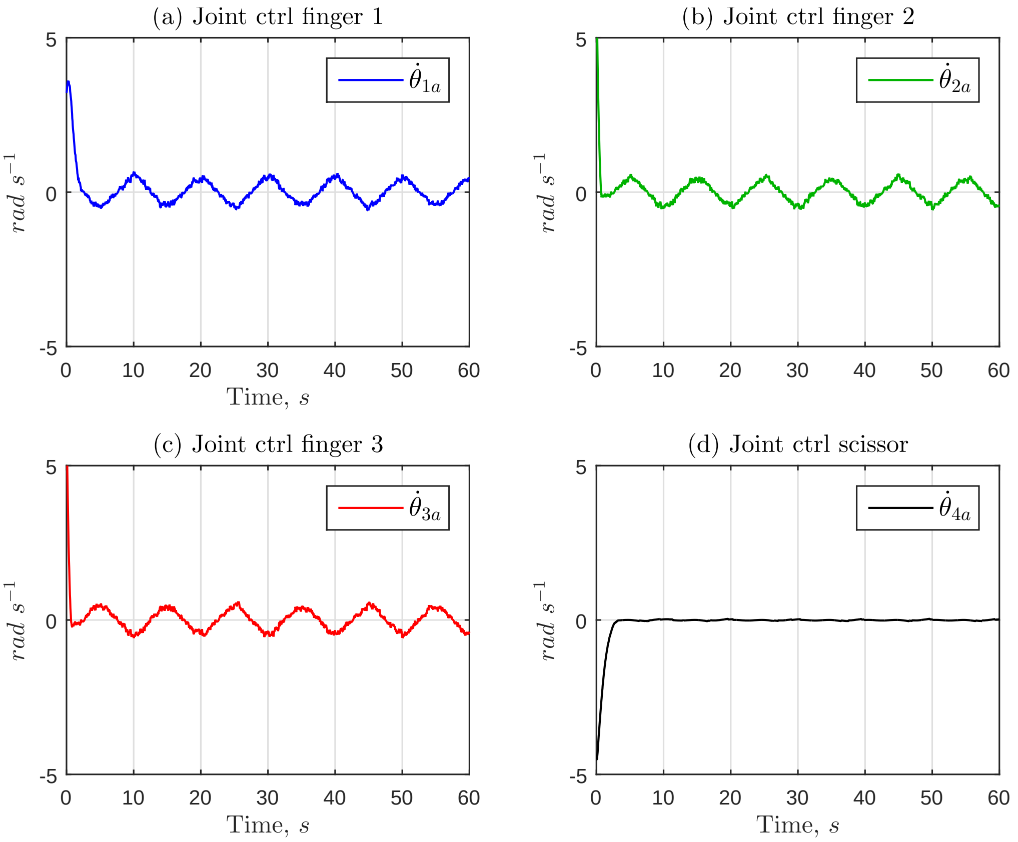
1. Reference and finger positions for the object tracking experiment.



2. Error signals for the object tracking experiment.



3. Position control signals for the object tracking experiment.



4. Joint control signals for the object tracking experiment.

Figure 4.7: Results for the object tracking experiment.

From the analysis of the results, we conclude that the proposed control scheme was able to ensure the asymptotic convergence of the object position error and of the relative position error to zero in approximately $3.2\,s$. From 4.7, we notice that the object tracking error converges to a small residual of less than $3\,mm$ in absolute value, due to the system delay, and the relative position converges to the desired values, ensuring the satisfactory prehension of the object during the whole interval of the experiment. The joint control signals remain bounded to less than $1\,rad\,s^{-1}$ of magnitude. These results demonstrate the non-occurrence of singular configurations in the task space of the mechanism, as expected from the analysis of the analytical expression of the system Jacobian.

4.2.4 HVS Control Experiments

In this section, we present two tracking experiments using the HVS approach. The first consists in controlling the centroid of the object's projected image in the *x-axis* of the camera. The second experiment consists in controlling the depth of the object by directly measuring it using the Kinect's depth sensor.

In these experiments, we have used a plastic red ball as the manipulated object. As before, an *ad-hoc* Matlab routine closes the fingers around the ball before running the vision-based control loop. The relative positions of the fingers are obtained by the forward kinematics map, and thus, the desired relative positions are set. For the the experiments of this section:

$$\xi_{rd} = S_r \, p_{rd} \approx [\, -75.1 \;\; 30.4 \;\; -1.9 \,]^{\mathsf{T}} \; mm \, .$$

For the sake of proof-of-concept and simplicity, we consider that there is no relative motion between the palm and camera frames, so that the control design developed in section 3.3.4 is valid.

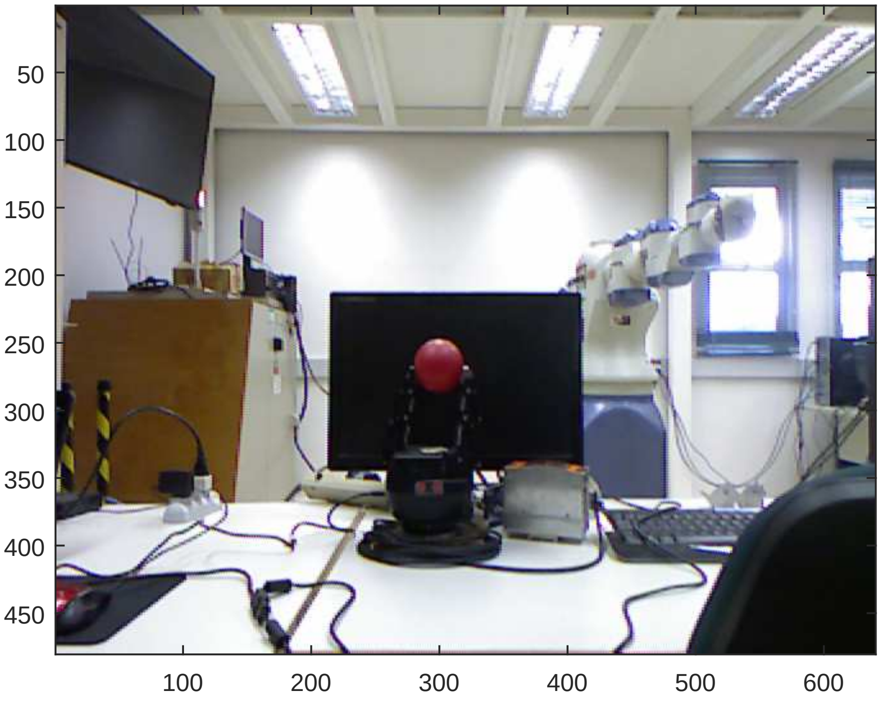


Figure 4.8: Setup for planar HVS experiment.

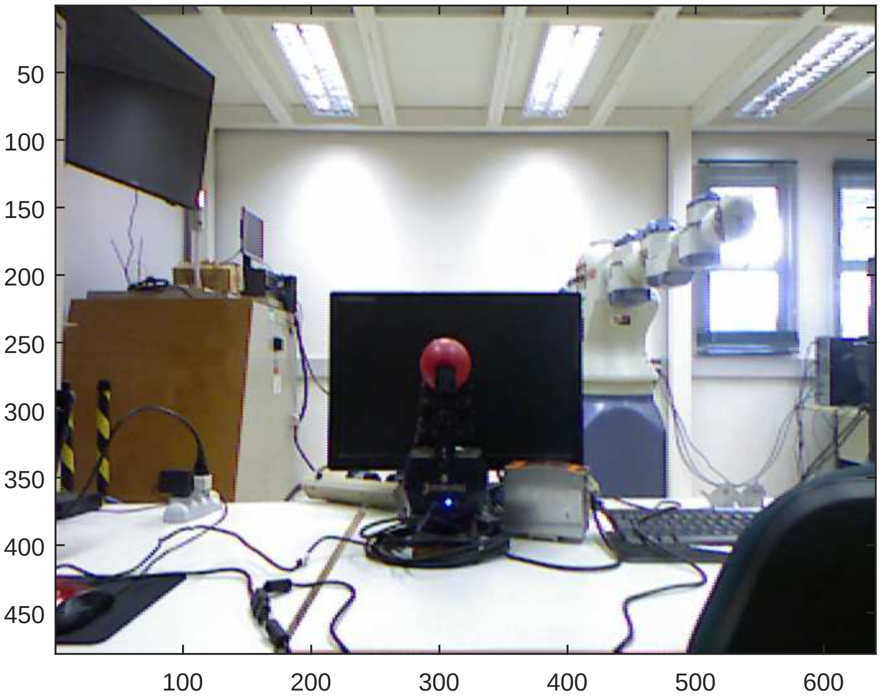


Figure 4.9: Setup for depth HVS experiment.

The rotation matrix R_{cp} between the camera and the palm frames is fixed:

$$R_{cp}^{(planar)} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix} \qquad R_{cp}^{(depth)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix},$$

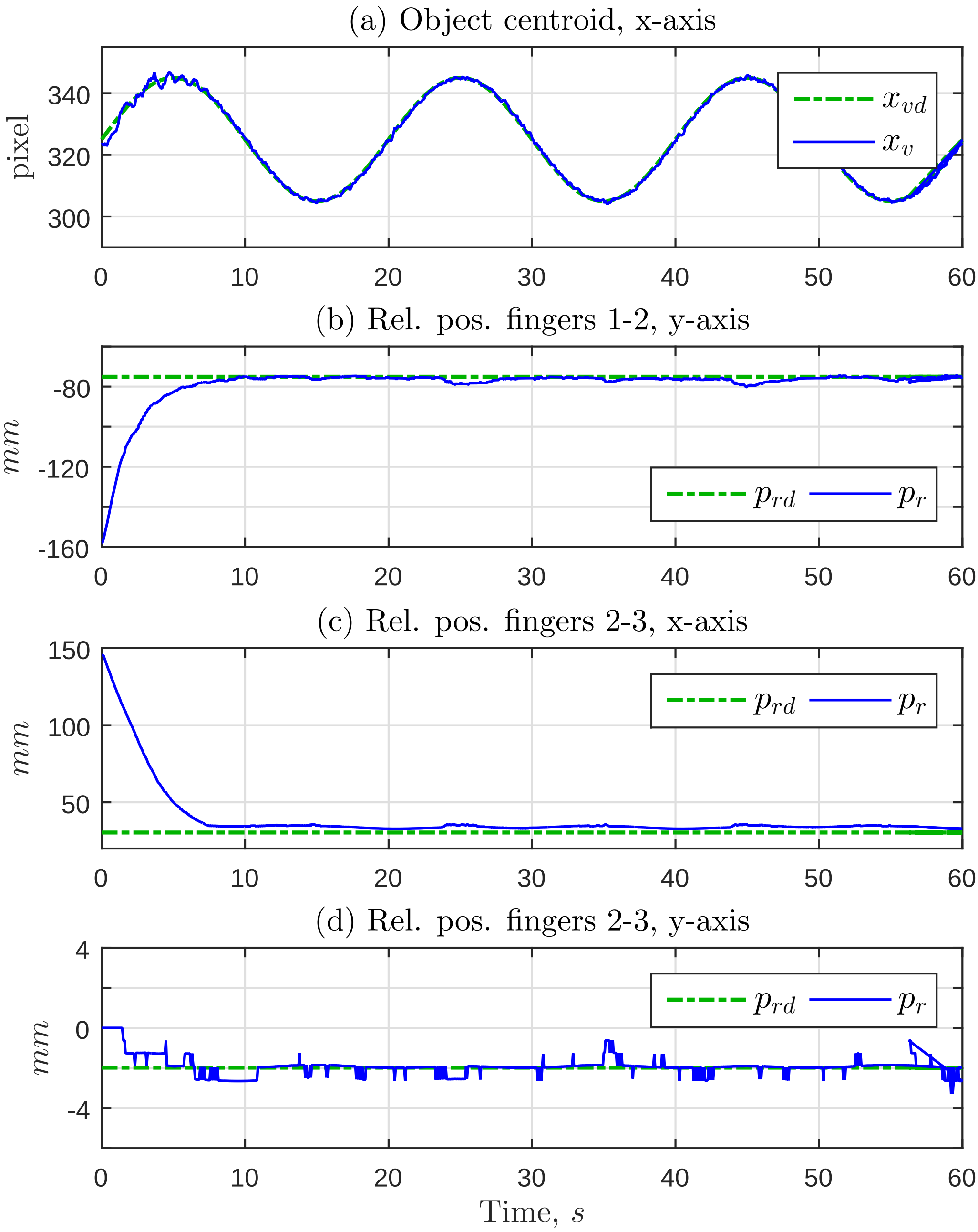
according to Figs. 4.8 - 4.9. The Microsoft Kinect is mounted on a tripod suitably positioned in front of the Robotiq gripper, so that:

1. For the centroid experiment, the image plane of the RGB camera is parallel to the motion plane of the grasped object (i.e, the object performs only *planar motions* on the image space) - Fig. 4.8; ;

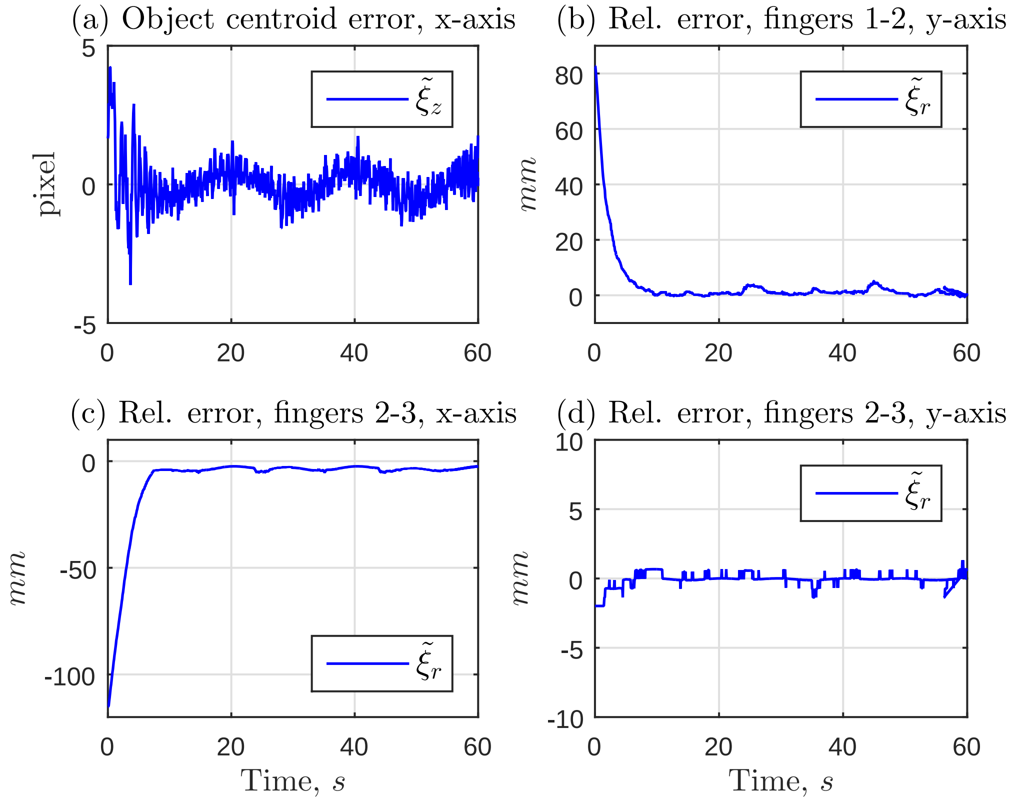
2. For the depth experiment, the image plane of the RGB camera is perpendicular to the motion plane of the grasped object, and both planes are perpendicular to the ground plane - Fig. 4.9 .

The scalar gains are set up to $\lambda_z = 10\, s^{-1}$ and $\lambda_r = 1\, s^{-1}$, where the gain matrices are defined as in the previous section, with $\lambda_z = \lambda_o$, $\Lambda_z = \Lambda_o$.

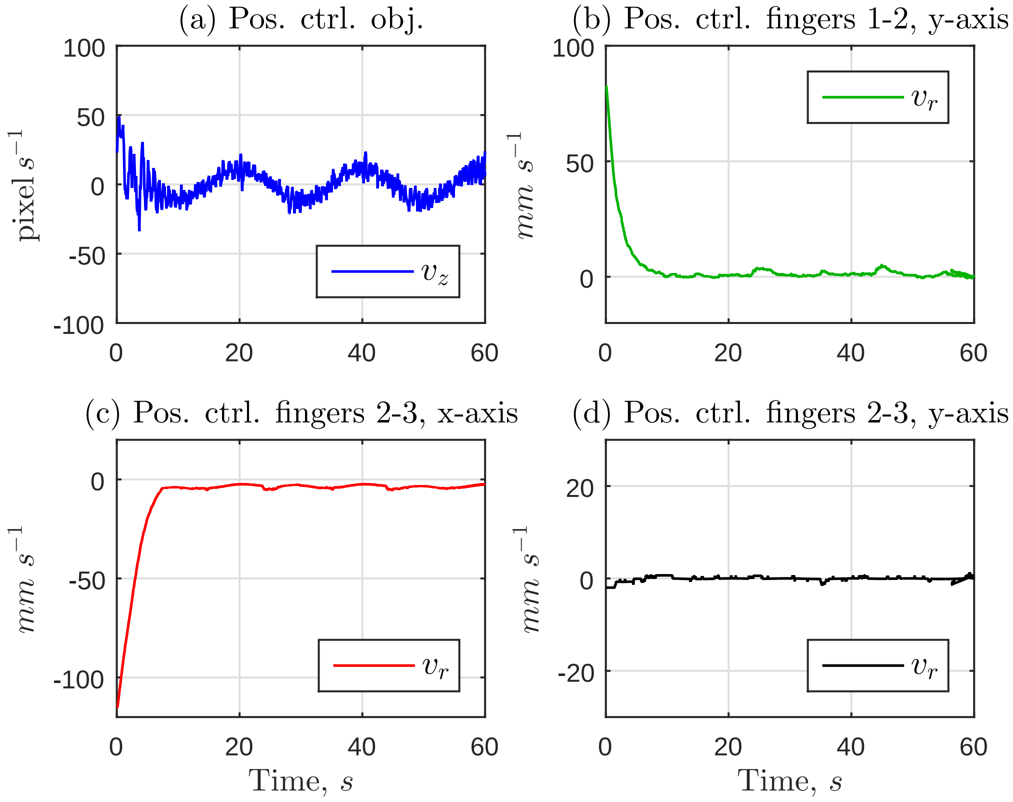
The results for the planar experiments are shown in Fig. 4.10. The measured depth remains constant during the hole interval of the experiment, since the motion plane is parallel to the camera frame. The parameters for the reference sinusoid are set up to $a = 20\,pixel$, $\omega = 0.1\,rad\,s^{-1}$ and $b = 325\,pixel$.



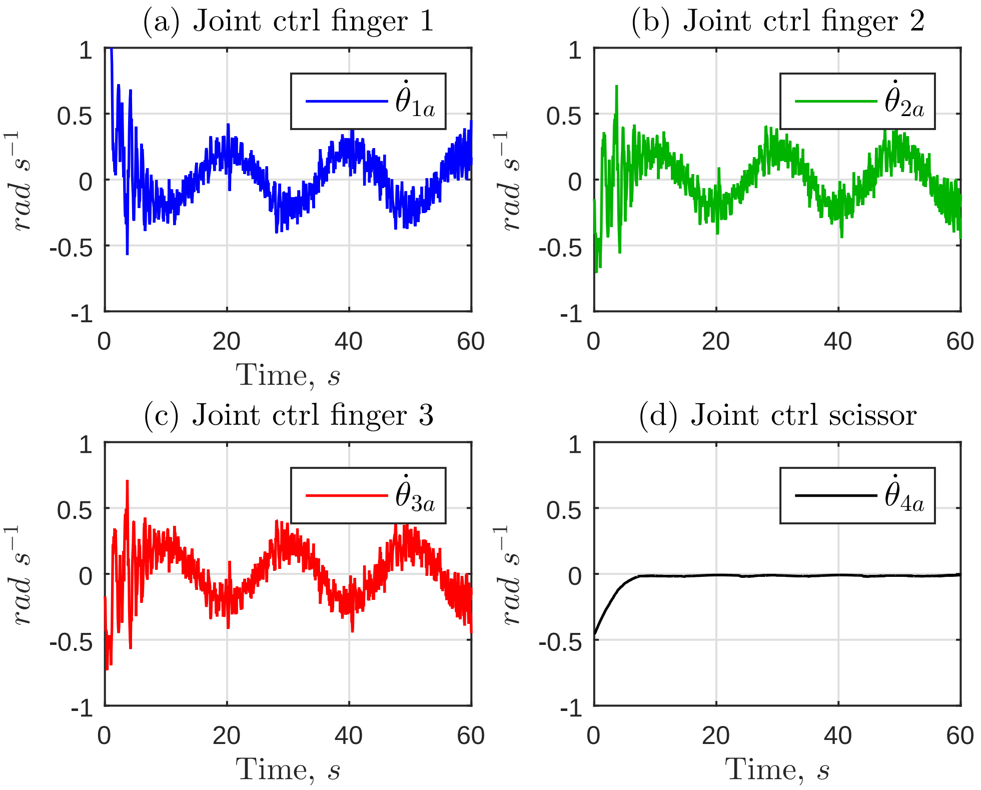
1. Reference and HVS state vector for planar experiment.



2. Error signals for planar HVS experiment.



3. Position control signals for planar HVS experiment.

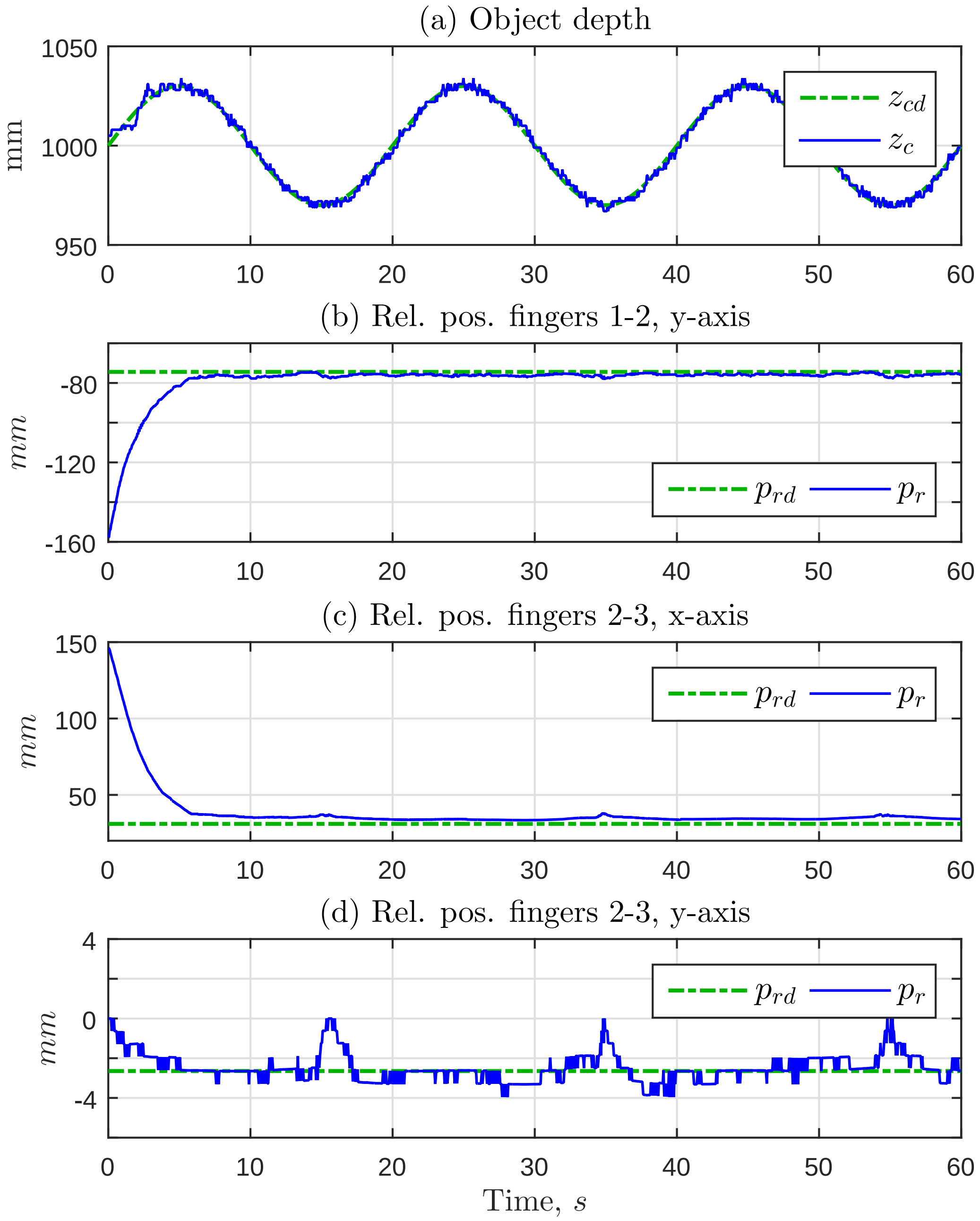


4. Joint control signals for planar HVS experiment.

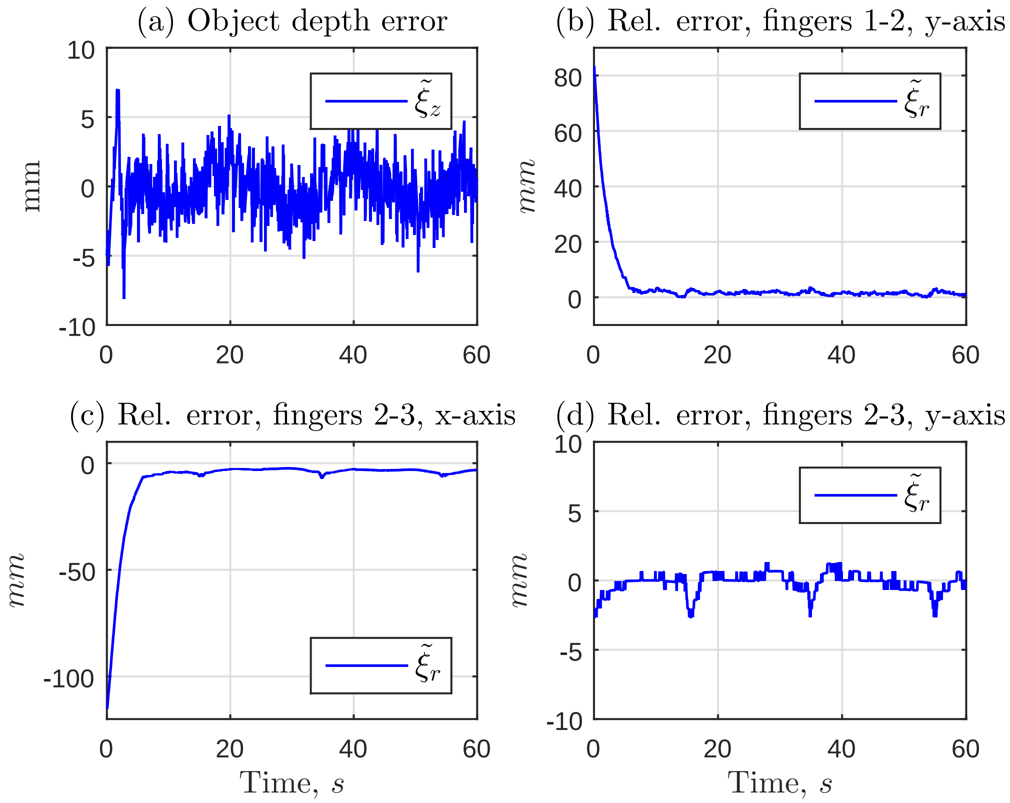
Figure 4.10: Results for the planar HVS experiment.

From these results, we conclude that, besides the relatively high signal-to-noise ratio of the Kinect's depth measurements, convergence of the object's image centroid is obtained, even in the presence of initial disturbances due to the slow convergence of the relative position. Of course, the object is not regrasped until the relative position error is small enough, but remains fixed on the previous grasp position. This is the reason why the convergence of the object image centroid is not fully achieved until the object is finally regrasped; because the image system is an open-control loop until it happens. Although the object prehension was maintained during the whole experiment, we note that there is a small offset error in the x *coordinate* between fingers 2 and 3 (Fig. 4.10.2-c). This offset error occurs due to a mechanical limit existing in the scissor motion of the Robotiq hand.

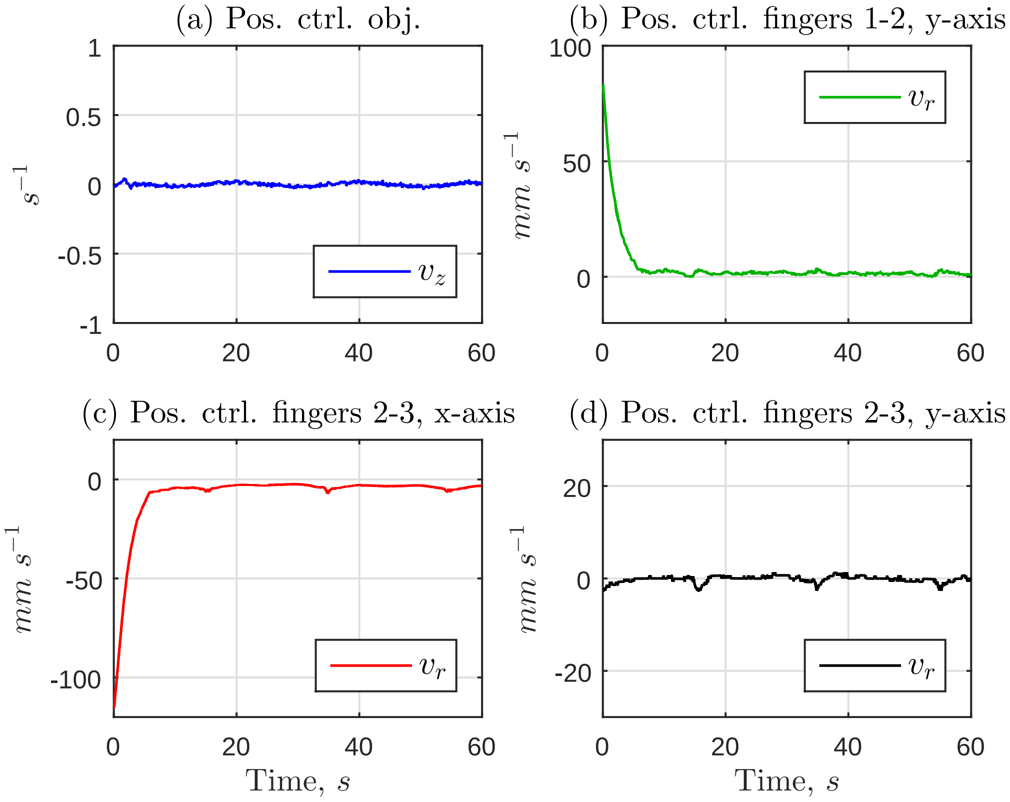
In the depth tracking experiment, the depth is continuously measured by the Kinect's depth sensor. A routine selects a small rectangle of the point cloud and estimates the depth of the ball by applying the *mean* to all depth points inside the rectangle. The results are shown in Fig. 4.11. The parameters for the reference sinusoid are set up to $a = 30\,mm$, $\omega = 0.1\,rad\,s^{-1}$ and $b = 1000\,mm$.



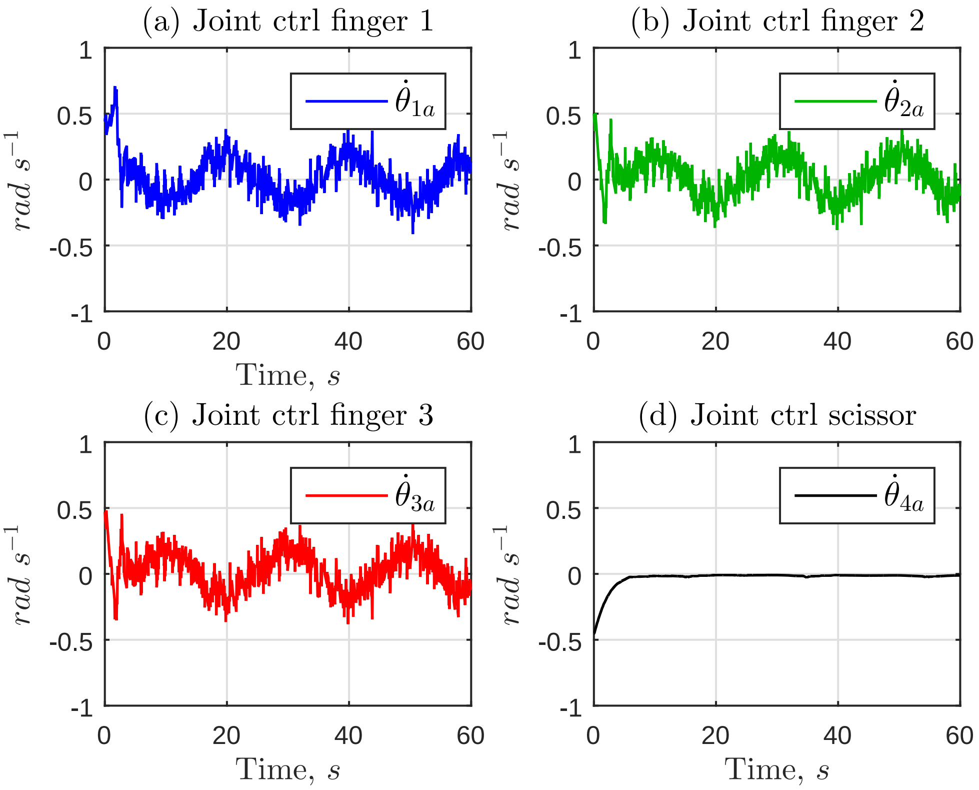
1. Reference and HVS state vector for depth experiment.



2. Error signals for depth HVS experiment.



3. Position control signals for depth HVS experiment.



4. Joint control signals for depth HVS experiment.

Figure 4.11: Results for the depth HVS experiment.

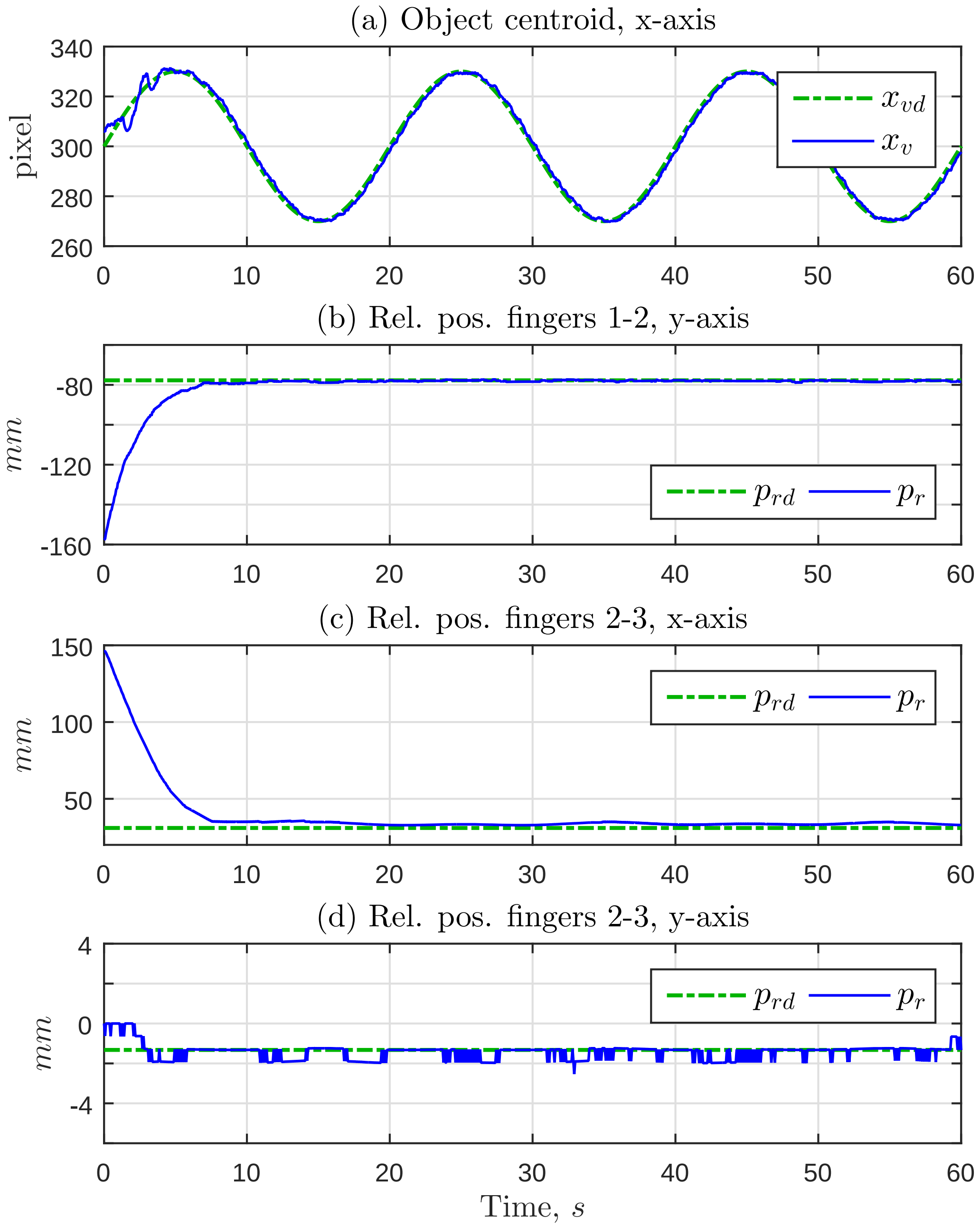
From the analysis of the results, we conclude that the HVS control scheme was able to ensure the asymptotic convergence of the logarithmic depth error to zero, as long as of the relative position error, even with a higher signal-to-noise ratio for the depth measurements. The logarithmic function has the interesting property of *shrinking* the measured depth signal, which potentially reduces the measurement noise. From Figs. 4.11.1-b, 4.11.1-c and 4.11.1-d, we can notice small peaks of approximately $3\,mm$ of magnitude in the relative position errors precisely in the time instants were the reference reaches its minimum value. This could be due to a large amplitude for the depth sinusoid; the minimum values of the depth reference could be leading some fingers to the limits of their task spaces. Besides, the depth error and the joint control signals remain bounded in the intervals $[-5,\, 5]\,mm$ and $[-0.5,\, 0.5]\,rad\,s^{-1}$, respectively, and the object prehension was maintained during the whole during of the experiment. The approximate system delay is about $0.05\,s$, for both experiments. These experimental results demonstrate the non-occurrences of singular configurations on the HVS image Jacobian and in the system Jacobian as a whole, considering the operation restricted to the image task space.

4.2.5 IBVS Control Experiments

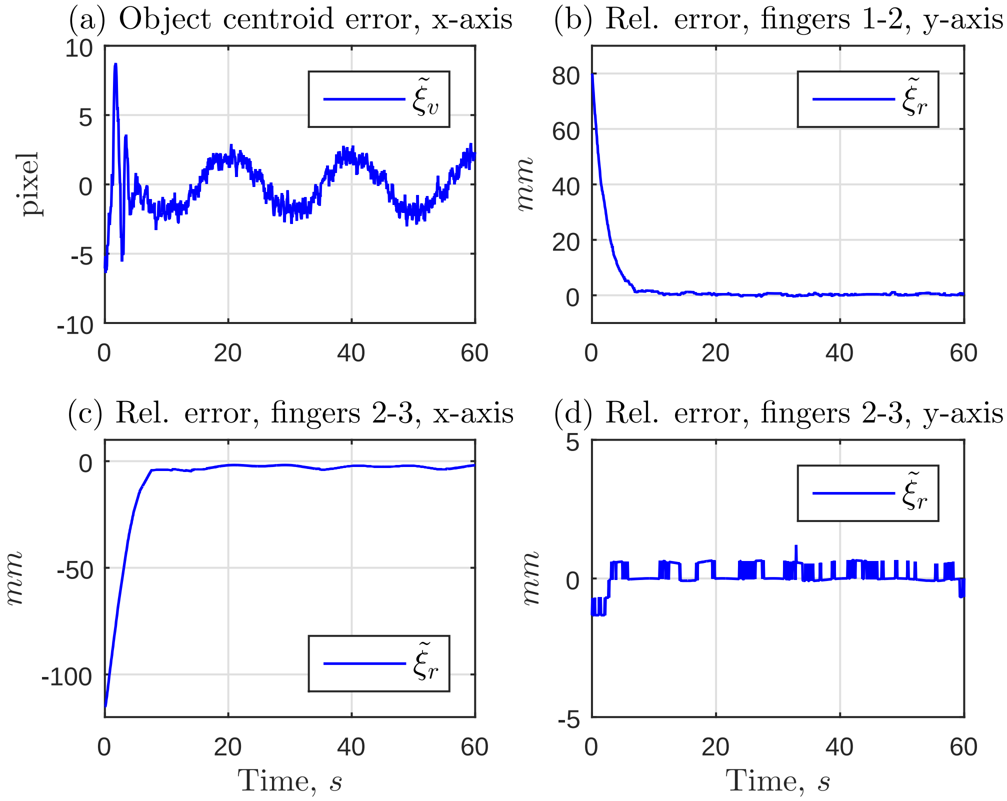
In this section, we present two tracking experiments using the IBVS approach. The first experiment is equivalent to the HVS planar experiment presented in the last section. Once again, the objective is to control the centroid of the object's projected image in the *x-axis* of the camera, but this time using the IBVS approach. The second experiment consists in controlling the image projected area of the object. The image projected area is obtained by counting the colored pixels corresponding to the object image on the RGB camera of the Kinect sensor, and can be used as an estimate of the object depth by means of (3.42) from section 3.3.5.

The same assumptions and observations made in the previous section about the control system are valid in this section, and the two experiments are performed in a similar way. A notable difference is that, in the IBVS depth experiment, the relative distance between the camera and the palm frame can be much smaller than in the HVS depth experiment, since we are not restricted to the minimum distance of 800 mm needed to obtain depth measurements with the Kinect's depth sensor. This increases the accuracy of this method, since the depth estimation by pixel count is more accurate when more pixels are available for counting, i.e, when the object is relatively close to the camera.

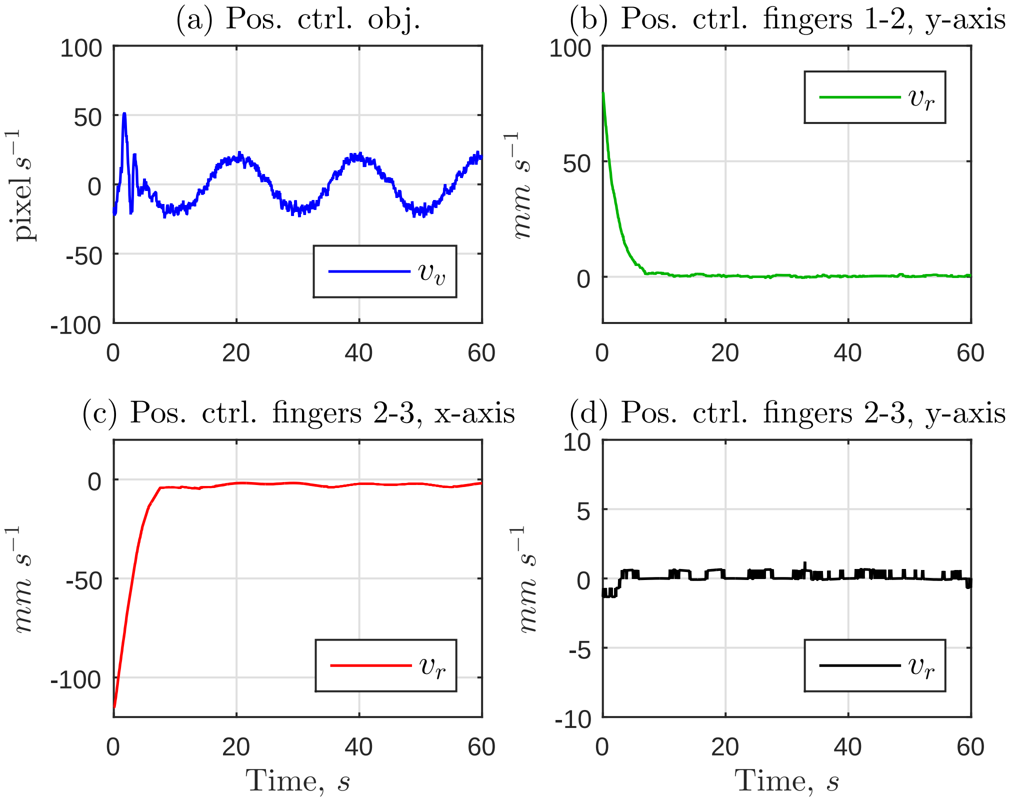
The scalar gains are set up to $\lambda_v = 5\, s^{-1}$ and $\lambda_r = 1\, s^{-1}$, where the gain matrices are defined as in the previous section, with $\lambda_v = \lambda_z$, $\Lambda_v = \Lambda_z$. The results are shown in Fig. 4.12. The measured area remains approximately constant during the hole interval of the experiment, since the motion plane is parallel to the camera frame. The parameters for the reference sinusoid are set up to $a = 30\, pixel$, $\omega = 0.1\, \pi\, rad\, s^{-1}$ and $b = 325\, pixel$.



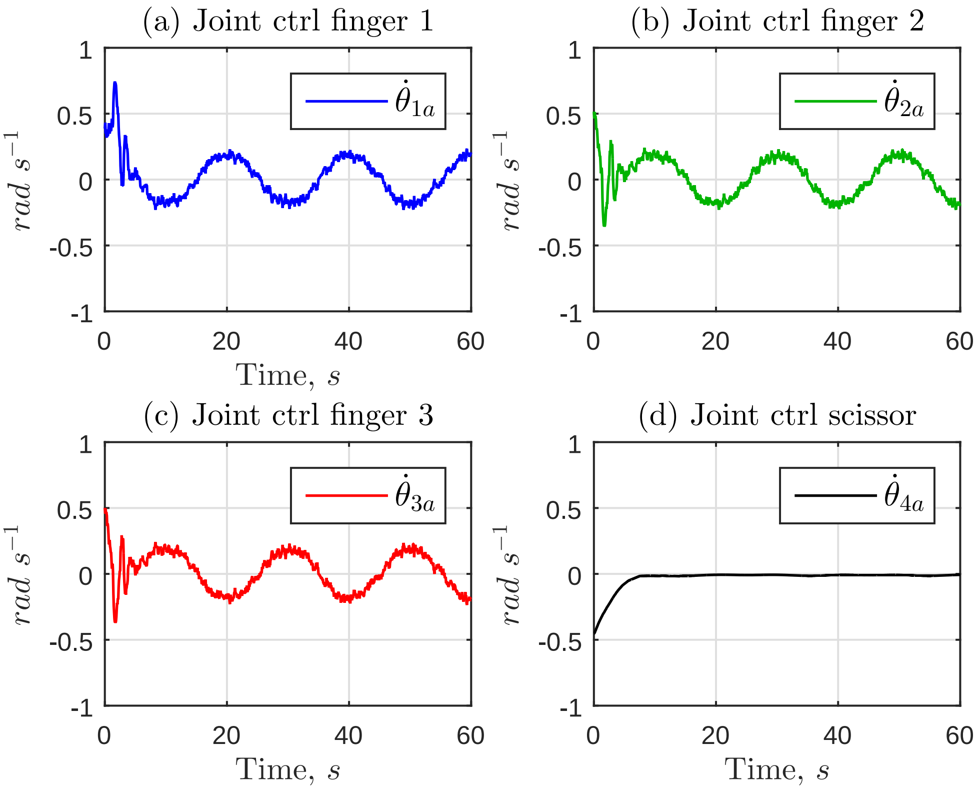
1. Reference and IBVS state vector for planar experiment.



2. Error signals for planar IBVS experiment.



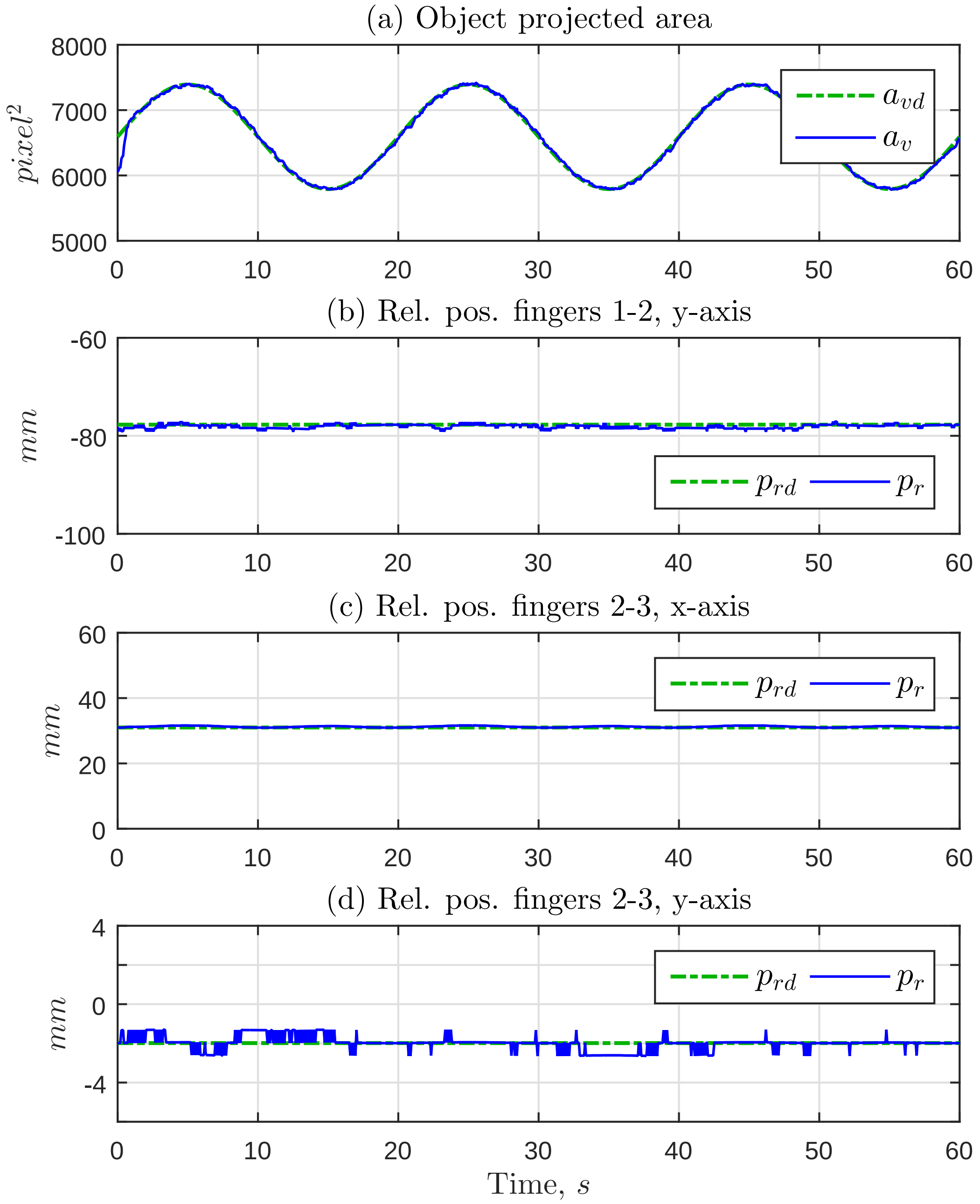
3. Position control signals for planar IBVS experiment.



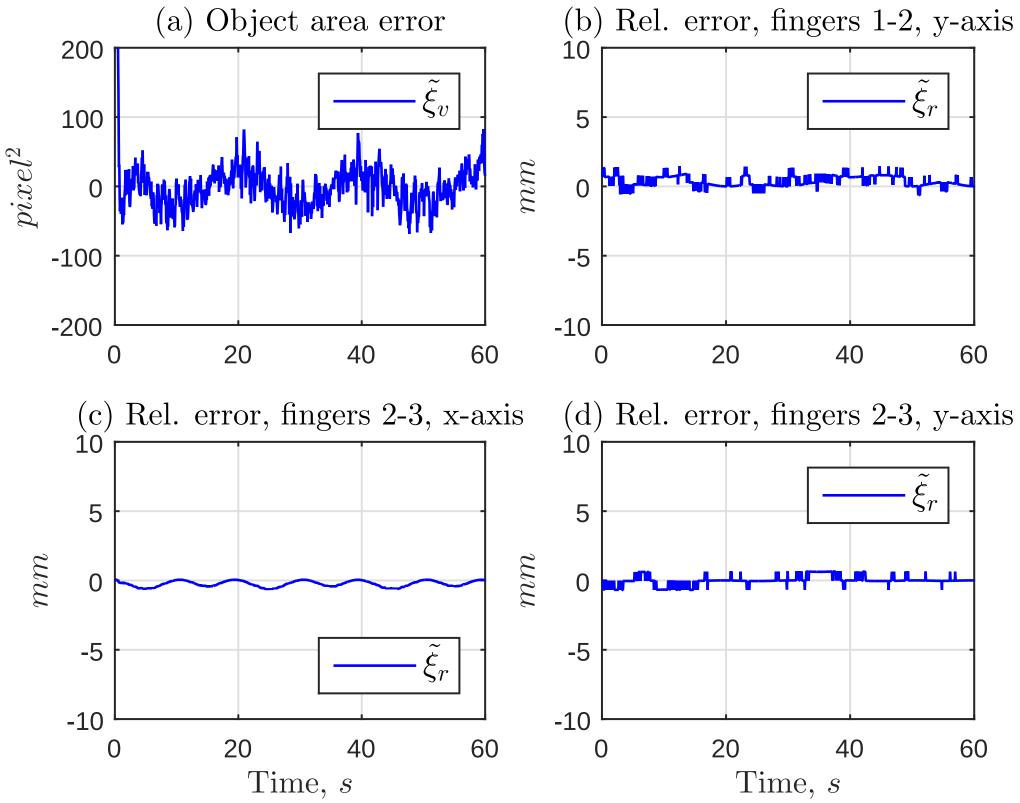
4. Joint control signals for planar IBVS experiment.

Figure 4.12: Results for the planar IBVS experiment.

In the area tracking experiment, the projected area is continuously measured by a proper color extraction routine. The reference is directly given in $pixel^2$, but equation (3.42) could be used to convert the desired depth in the corresponding projected area. The results are shown in Fig. 4.13. The parameters for the reference sinusoid are set up to $a = 800\,pixel^2$, $\omega = \frac{\pi}{10}\,rad\,s^{-1}$ and $b = 6500\,pixel^2$.



1. Reference and IBVS state vector for depth experiment.

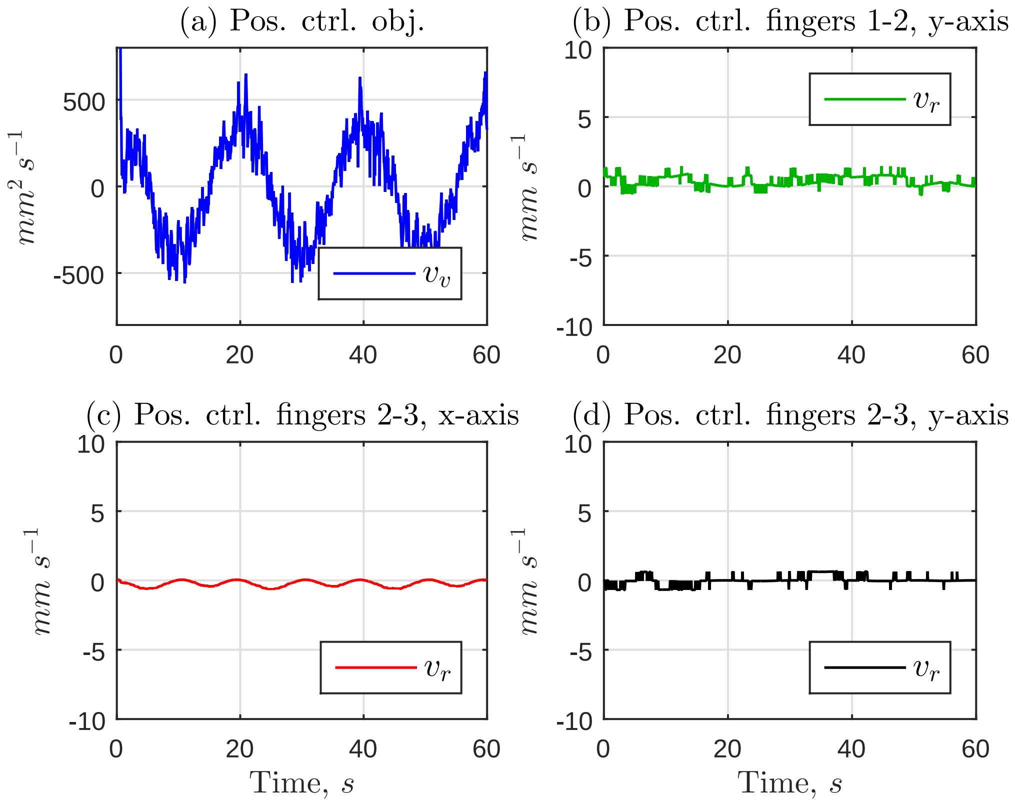

(a) Object area error

(b) Rel. error, fingers 1-2, y-axis

(c) Rel. error, fingers 2-3, x-axis

(d) Rel. error, fingers 2-3, y-axis

2. Error signals for depth IBVS experiment.

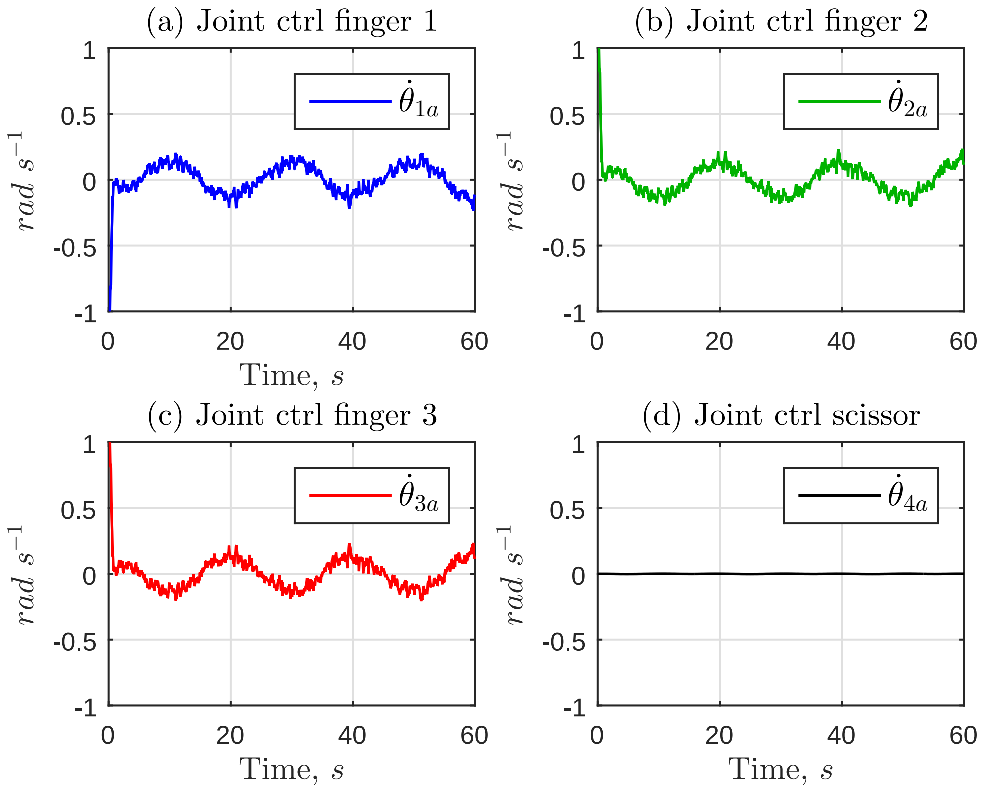

(a) Pos. ctrl. obj.

(b) Pos. ctrl. fingers 1-2, y-axis

(c) Pos. ctrl. fingers 2-3, x-axis

(d) Pos. ctrl. fingers 2-3, y-axis

3. Position control signals for depth IBVS experiment.



4. Joint control signals for depth IBVS experiment.

Figure 4.13: Results for the depth IBVS experiment.

From the analysis of the results, we conclude that the IBVS control scheme is also able to ensure the asymptotic convergence of the image error, for both experiments. The small peaks in the relative position errors observed in the HVS depth experiment are also observed here, and the reason is probably the same. It is important to notice that the IBVS method is apparently less sensitive to measurement noise than the HVS method. The joint control signals remain bounded in the interval $[-0.25, 0.25]\,rad/s$, The planar error signal is less than $3\,pixel$ in magnitude, and the area error is bounded to $[-100, 100]\,pixel^2$. The approximate system delay is the same as before, about $0.05\,s$. These experimental results demonstrate the non-occurrences of singular configurations on the IBVS image Jacobian and in the system Jacobian as a whole, considering the operation restricted to the image task space.

4.2.6 Simulation of a 12 Dof Robot Hand

In this section, we present the results of the kinematic control of a simulated three-fingered robot hand with 12 degrees-of-freedom, 4 per finger. Its structure is similar to the Robotiq gripper without the finger constraints and with an extra joint in the first finger. We suppose that each finger angle velocity can be directly controlled by means of an internal velocity control loop, so that the kinematic control approach can be applied to the problem. The extra joint, θ_{14} can be thought of a generalization of the "scissor" joints of fingers 2 and 3, as seen in Fig. 2.1 in Chapter 2. This joint allows finger 1 to perform motion in the x-axis of the palm frame.

The forward kinematics map for the first finger is modified:

$$p_1 = p_{1_{offset}} + \underbrace{\begin{bmatrix} (l_c^1 + l_0)\, s_4^1 \\ l_s^1 \\ (l_c^1 + l_0)\, c_4^1 \end{bmatrix}}_{p_{1_{link}}} \quad , \quad R_1 = \begin{bmatrix} c_4^1 & -s_{123}^1\, s_4^1 & c_{123}^1\, s_4^1 \\ 0 & c_{123}^1 & s_{123}^1 \\ -s_4^1 & -s_{123}^1\, c_4^1 & c_{123}^1\, c_4^1 \end{bmatrix} .$$

Note that if $\theta_{14} = 0$, it reduces to the forward kinematics of the Robotiq gripper in (2.39). The vector of finger angles θ_1 is modified as:

$$\theta_1 = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} & \theta_{14} \end{bmatrix}^{\mathsf{T}} ,$$

and also the first finger Jacobian matrix J_f^1:

$$J_f^1 = \begin{bmatrix} -l_s^1\, s_4^1 & -d_s^1\, s_4^1 & -l_3\, s_{123}^1\, s_4^1 & (l_c^1 + l0)\, c_4^1 \\ l_c^1 & d_c^1 & l_3\, c_{123} & 0 \\ -l_s^1\, c_4^1 & -d_s^1\, c_4^1 & -l_3\, s_{123}^1\, c_4^1 & -(l_c^1 + l0)\, s_4^1 \\ -c_4^1 & -c_4^1 & -c_4^1 & 0 \\ 0 & 0 & 0 & 1 \\ s_4^1 & s_4^1 & s_4^1 & 0 \end{bmatrix} .$$

Again, it reduces to the Jacobian matrix of finger 1 of the Robotiq hand when $\theta_{1s} = 0$.

Finally, since there are no finger kinematic relations between the joint angles and no shared angles, the expression for the hand Jacobian matrices simplifies to:

$$J_h = \begin{bmatrix} J_f^1 & 0 & 0 \\ 0 & J_f^2 & 0 \\ 0 & 0 & J_f^3 \end{bmatrix} , \qquad J_{Ph} = \begin{bmatrix} J_{Pf}^1 & 0 & 0 \\ 0 & J_{Pf}^2 & 0 \\ 0 & 0 & J_{Pf}^3 \end{bmatrix} ,$$

where J_{Pf}^k, $k = 1, 2, 3$ are the position parts of the finger Jacobian matrices J_f^k. With these expressions at hand, it is possible to develop kinematic control algorithms for object manipulation tasks.

From section 3.3.3, the state vector is defined as:

$$\xi = \begin{bmatrix} p_o \\ p_r \end{bmatrix}$$

with the vector of relative positions p_r defined the same way as before. It means that all selection matrices are identities, and thus, all coordinates are controlled in this case (the state vector is complete).

A Matlab routine first uses a kinematic control algorithm to lead the fingers to fixed contact points in the simulated object, in the case, a ball of radius $r = 30\,mm$. Then, it applies the kinematic control algorithm detailed at section 3.3.3 to perform a position tracking experiment with the grasped object.

Here, the rotation matrix R_{po} is calculated at each interaction, using J_{Oo} (the orientation part of the object Jacobian matrix) to estimate the angular velocity of the object. Then, applying *Quaternion transformations*, it is possible to integrate $\dot{R}_{po}$, obtaining a new rotation matrix for the next iteration. Although the object orientation problem is not covered by this work, this approach works relatively well for online estimation of R_{po}.

Fig. 4.3 shows the simulation environment, the grasped object and the robot hand. The reference for the object position is given by:

$$p_{od} = \begin{bmatrix} a_x\,sin(w\,t) + b_x & a_y\,sin(w\,t) + b_y & a_z\,sin(w\,t) + b_z \end{bmatrix}^{\mathbf{T}}$$

where $a_x = a_y = a_z = 20mm$, $b_x = b_y = 0$, $b_z = 50$ and $w = \frac{2\pi}{5}$ (the parametric expression of an ellipse in the task space). The initial conditions for the simulation are $R_{po} = I_3$ and $p_o = [0\;\;0\;\;50]^{\mathbf{T}}$.

Fig. 4.14 illustrates the task space trajectory of the object center and Fig. 4.15 illustrates the object position, errors and joint control signals. The relative positions and the position control signals are ommited for the sake of simplicity.

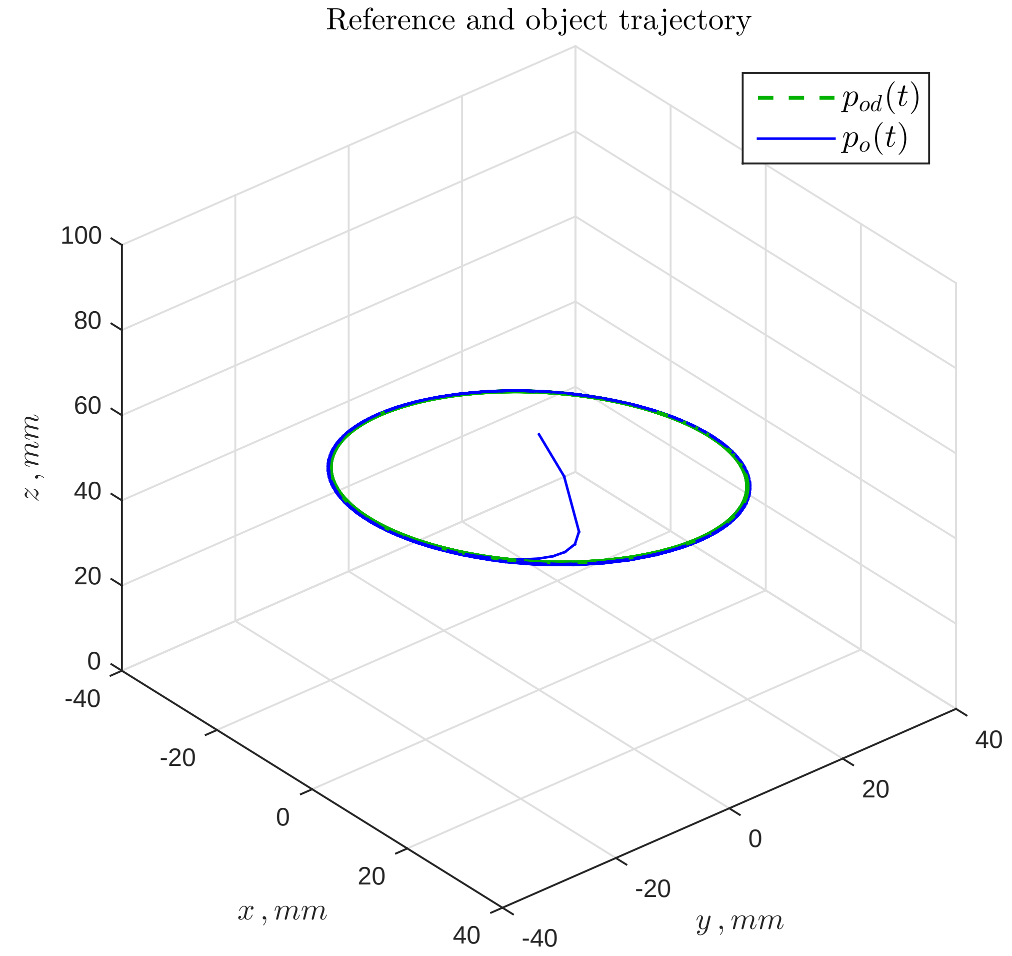
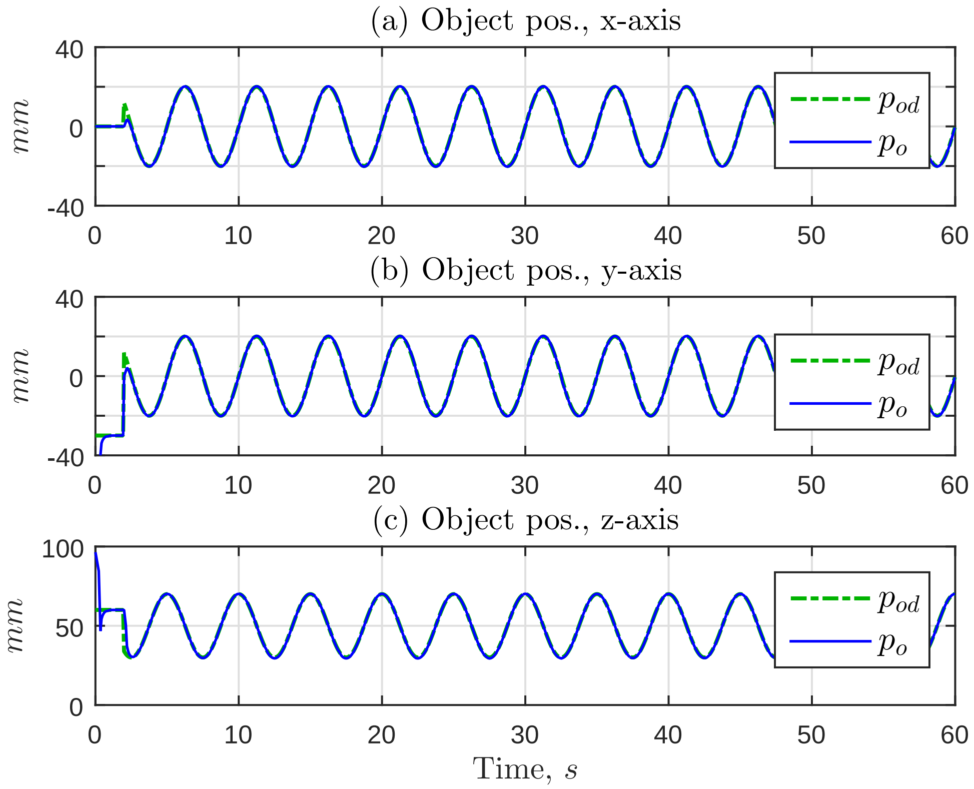


Figure 4.14: Reference and object trajectory for the tracking experiment.

These results demonstrate the asymptotic convergence of the object tracking and relative position errors to zero by the kinematic control approach for object manipulation developed in section 3.3.3. The magnitude of the tracking error is bounded to a small residual of $1\,mm$, after the convergence of the relative position states. This is due to the small simulation sample time of $0.067\,s$, much smaller than the sample time of the experiments with the Robotiq hand, because of limitations in comunication speed imposed by the USB connection. The magnitude of the joint control signals also remains bounded to less than $1\,rad\,s^{-1}$, an acceptable joint velocity range for a real 12 Dof robot hand.

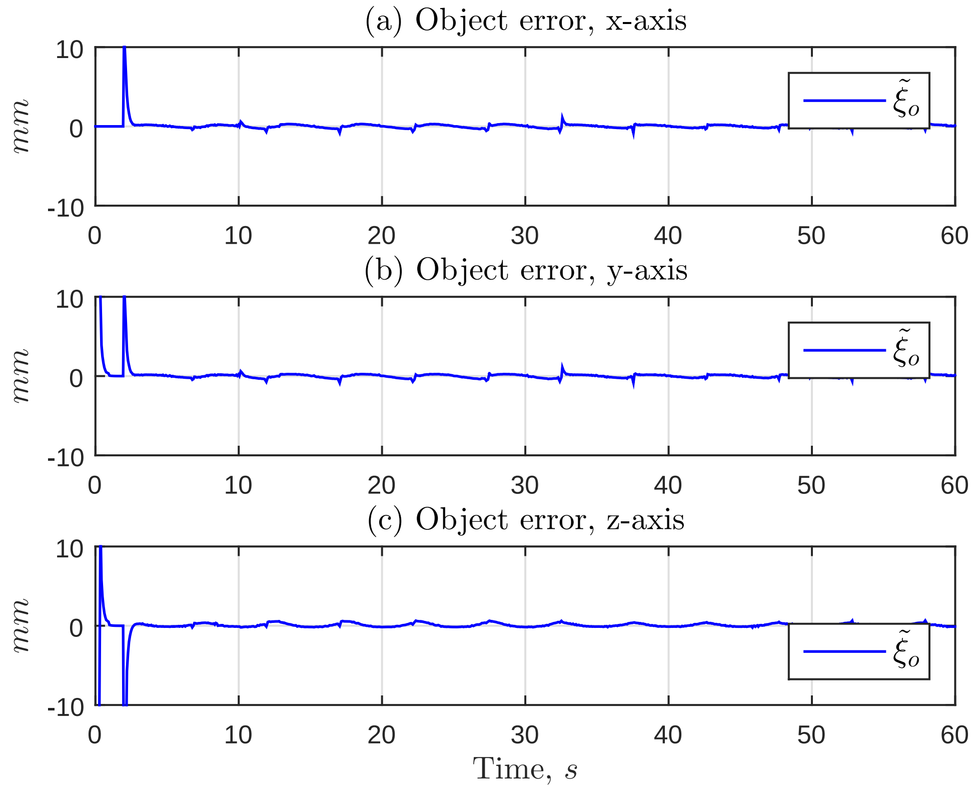


(a) Object pos., x-axis

(b) Object pos., y-axis

(c) Object pos., z-axis

1. Reference and object position.

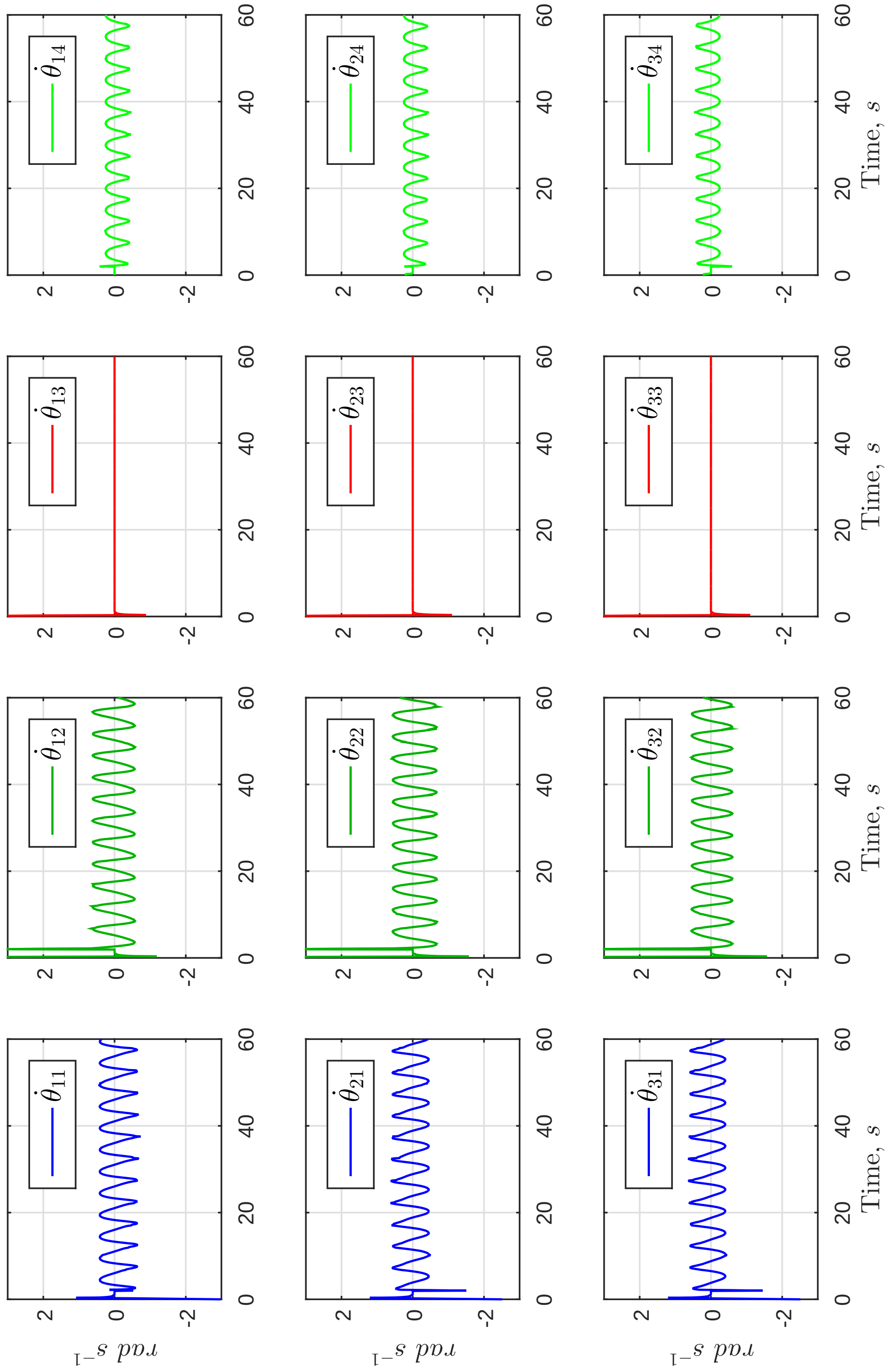


(a) Object error, x-axis

(b) Object error, y-axis

(c) Object error, z-axis

2. Object position error.



3. Joint control signals.

Figure 4.15: Results for tracking experiment with a 12 Dof robot hand.

4.3 Analysis of Results

Since friction effects were not taken into account in the proposed methodology, the kinematic control approach without explicit closure of a force feedback loop is not able to guarantee the satisfactory prehension of the object during the manipulation task. To overcome this drawback, we have combined the object position control with the relative position control for the fingers, ensuring a firm grasp of the object. Although, this scheme for object prehension works better if the object or the fingertips are made of soft materials. For example, in sections 4.2.3, 4.2.4, 4.2.5, we have discussed the presence of small errors in the relative positions that could have lead to grasping fail, if the grasped object was completely rigid. The nature of the contact surface is also very important: since no force-control loop is applied, the performance of the fingertip grasp relies on the friction coefficient of the fingertip contact, which depends on the material of the object.

With respect to the visual servoing schemes, besides their great advantages pointed in Chapter 1, some limitations may arise, since these approaches require:

- (i) CPUs with higher clock speed to perform the necessary calculations and execute image processing algorithms, mainly when the motions of the target object are very fast;

- (ii) high resolution cameras to provide more accurate pixel counting for the image projected area and allow the execution of visual servoing tasks for greater distances.

An important remark to be done is about the main differences between the two approaches for visual servoing noticed by the observation of the experiments. The first of them is the signal-to-noise ratio. Even by attenuating the measured depth signal by applying the natural logarithm, in general, the results of the HVS method are more noisy than the IBVS. Of course, it is clear that the IBVS performance for depth control *will decay* if the Robotiq hand is positioned too far from the camera, while the performance with the HVS method will remain unaltered. This is due to the nature of the method used for depth estimation for each scheme. In the HVS method, the depth is directly measured by the Kinect's depth sensor, being affected by the point cloud noise and by the range of measurement only. In the IBVS method, the depth is estimated by pixel counting, a method that is more accurate if there are a great number of pixels in the projected image of the object, i.e, if the object is relatively close to the Kinect. If this is not the case, the resolution limitation of the RGB camera starts to affect the performance, as pointed out in (ii).

With respect to the eigenvalues of the gain matrices, they have been carefully limited for the sake of stability and convergence of the control algorithms. Although high values of control gains increase the dynamic response of the closed-loop system, and thus, the overall performance, they can also lead the system to instability, because of communication limitations that impose finite inferior limits to the sample time of the algorithm.

The simulation results of a 12 Dof robot hand performing an object manipulation task helped to visualize the applicability and effectiveness of the framework of cooperative robotics applied to more complicate problems, such as the control of complex robot hands with many degrees-of-freedom. This is not the case of the Robotiq hand, because of its mechanical limitations and few degrees of freedom. However, it demonstrates that the methodology can be applyed in many advanced prototypes of multifingered robot hands with rigid fingers, such as Schunk (LIU *et al.*, 2007), Barret (TOWNSEND, 2000) and even the Shadow hand (SHADOW ROBOT COMPANY, 2003).

Finally, the theoretical results of this work hold in the practical experiments under the assumptions of full knowledge of the model and camera parameters, and no occurrences of singular configurations. The kinematic control approach turned out to be very applicable to control the Robotiq hand, because of its high gear ratios and slow finger motions. A force control scheme was not utilized because of the lack of a precise force measuring device in the Robotiq hand, such as a tactile sensors in the fingertips or precise current sensors in the driver motors, a difficulty that has been overcome by controlling the relative finger position.

Chapter 5

Conclusions and Future Works

In this chapter, we present the conclusions about this work and discuss some perspectives for future works in the area.

5.1 Conclusion

In this work, we have presented a kinematic framework for modeling and control design of a multifingered robot hand, which is based on an empirical model and a control formalism developed for general robotic systems subject to kinematic constraint conditions (WEN e WILFINGER, 1999).

The position control problems for the fingers and the manipulated object were tackled by using the kinematic control approach, which ensures the asymptotic stability of the output tracking errors. Besides, we also have presented two visual servoing schemes to solve the problem of controlling the grasped object in the image space. Experimental results, obtained with the 3-Finger Adaptive Robot Gripper from Robotiq, were shown and discussed to illustrate the performance and effectiveness of the methodology.

We conclude by pointing out that, besides the general framework of cooperative robots and the kinematic control approach were effective for controlling the Robotiq hand and are applicable to a wide class of robot hands, they also have limitations. For example, consider the problem of non-prehensile grasping tasks. Although many advanced prototypes of robot hands have the required speed and manipulability to perform such complex tasks, the control schemes require the knowledge of the dynamic models of these devices, which are considerably more complex than the kinematic models considered in this work.

Finally, we point out that this work have originated three scientific papers, published at:

- ISIE 2015 (International Symposium in Industrial Electronics 2015)
  "Visual Servoing for Object Manipulation with a Multifingered Robot Hand" (REIS *et al.*, 2015);

- Syroco 2015 (11th IFAC Symposium on Robot Control)
  "Kinematic modeling and control design of a multifingered robot hand" (REIS *et al.*, 2015);

- CDC 2015 (IEEE 54th Annual Conference on Decision and Control)
  "Modeling and control of a multifingered robot hand for object grasping and manipulation tasks" (REIS *et al.*, 2015);

The work has also been awarded in the annual event named JIC (Giulio Massarani Scientific Initiation Journey) at Federal University of Rio de Janeiro in the year of 2014, as one of the ten works selected to receive an honorable mention award.

5.2 Future Works

Motivated by the questions discussed above, some topics for future developments are:

- (i) to implement hybrid position/force control strategies for object grasping and manipulation tasks for general objects, with different shapes and sizes;

- (ii) to develop intelligent grasping algorithms to select the best grasping for an object in terms of object manipulability and grasp stability;

- (iii) to perform path planning and object manipulation tasks with the Robotiq hand attached to a large-scale robot manipulator;

- (iv) to implement visual pose estimation methods for estimating the extrinsic parameters of the camera, allowing the visual servoing methods to be applied in the general case of relative motion between the palm and camera frames.

Bibliography

[1] PRATTICHIZZO, D., TRINKLE, J. C. “Grasping”. In: Siciliano, B., Khatib, O. (Eds.), *Springer Handbook of Robotics*, Springer Verlag, pp. 671–700, 2008. ISBN: 978-3-540-23957-4. doi: 10.1007/978-3-540-30301-5_29.

[2] BICCHI, A. “Hands for Dexterous Manipulation and Robust Grasping: a Difficult Road Toward Simplicity”, *IEEE Trans. Rob. Aut.*, v. 16, n. 6, pp. 652–662, Dec 2000. ISSN: 1042-296X. doi: 10.1109/70.897777.

[3] ALEOTTI, J., LODI RIZZINI, D., CASELLI, S. “Perception and Grasping of Object Parts from Active Robot Exploration”, *J. Intell. Rob. Syst.*, v. 76, n. 3-4, pp. 401–425, 2014. ISSN: 0921-0296. doi: 10.1007/s10846-014-0045-6.

[4] OKADA, T. “Object-Handling System for Manual Industry”, *IEEE Trans. Syst., Man and Cyber.*, v. 9, n. 2, pp. 79–89, Feb 1979. ISSN: 0018-9472. doi: 10.1109/TSMC.1979.4310152.

[5] MELCHIORRI, C., KANEKO, M. “Robot Hands”. In: Siciliano, B., Khatib, O. (Eds.), *Springer Handbook of Robotics*, Springer Verlag, pp. 345–360, 2008. ISBN: 978-3-540-23957-4. doi: 10.1007/978-3-540-30301-5_16.

[6] TOWNSEND, W. T. “The Barrett Hand Grasper - Programmably Flexible Part Handling and Assembly”, *Ind. Rob.: An. Int. J.*, v. 27, n. 3, pp. 181–188, 2000.

[7] SHADOW ROBOT COMPANY. “Design of a Dexterous Hand for Advanced CLAWAR Applications”. In: *Proc. 6th Int. Conf. CLAWAR*, pp. 691â€“–698, Nov 2003. ISBN: 978-1-86058-409-1.

[8] LIU, H., MEUSEL, P., SEITZ, N., et al. "The Modular Multisensory DLR-HIT-Hand", *Mech. Mach. Theory*, v. 42, n. 5, pp. 612–625, May 2007. ISSN: 0094-114X.

[9] HUTCHINSON, S., HAGER, G., CORKE, P. "A Tutorial on Visual Servo Control", *IEEE Transactions on Robotics and Automation*, v. 12, n. 5, pp. 651–670, Oct 1996. ISSN: 1042-296X. doi: 10.1109/70.538972.

[10] SICILIANO, B., SCIAVICCO, L., VILLANI, L., et al. *Robotics: Modelling, Planning and Control*. Springer-Verlag London, 2009.

[11] LUIGI VILLANI, J. D. S. "Force Control". In: Siciliano, B., Khatib, O. (Eds.), *Springer Handbook of Robotics*, Springer Berlin Heidelberg, pp. 161–185, 2008. ISBN: 978-3-540-23957-4.

[12] TINÓS, R., TERRA, M., ISHIHARA, J. "Motion and Force Control of Cooperative Robotic Manipulators with Passive Joints", *IEEE Trans. Contr. Syst. Tech.*, v. 14, n. 4, pp. 725–734, Jul 2006. ISSN: 1063-6536. doi: 10.1109/TCST.2006.872505.

[13] FREITAS, G. M., LEITE, A. C., LIZARRALDE, F. "Kinematic Control of Constrained Robotic Systems", *SBA Contr. Aut. Mag.*, v. 22, pp. 559–572, Dec 2011. http://dx.doi.org/10.1590/S0103-17592011000600002.

[14] KAO, I., LYNCH, K., BURDICK, J. W. "Contact Modeling and Manipulation". In: Siciliano, B., Khatib, O. (Eds.), *Springer Handbook of Robotics*, Springer Verlag, pp. 647–669, 2008. ISBN: 978-3-540-23957-4. doi: 10.1007/978-3-540-30301-5_28.

[15] YOSHIKAWA, T. "Multifingered Robot Hands: Control for Grasping and Manipulation", *Ann. Rev. Contr.*, v. 34, n. 2, pp. 199–208, 2010. ISSN: 1367-5788.

[16] TAKAHASHI, T., TSUBOI, T., KISHIDA, T., et al. "Adaptive Grasping by Multi fingered Hand with Tactile Sensor based on Robust Force and Position Control". In: *Proc. IEEE Int. Conf. Rob. Aut.*, pp. 264–271, May 2008. doi: 10.1109/ROBOT.2008.4543219.

[17] ENGEBERG, E., MEEK, S. "Adaptive Sliding Mode Control for Prosthetic Hands to Simultaneously Prevent Slip and Minimize Deformation of Grasped Objects", *IEEE/ASME Trans. Mech.*, v. 18, n. 1, pp. 376–385, Feb 2013. doi: 10.1109/TMECH.2011.2179061.

[18] YIN, Y., LUO, Z., SVININ, M., et al. "Hybrid Control of Multi-fingered Robot Hand for Dexterous Manipulation". In: *Proc. IEEE Int. Conf. Syst., Man and Cyber.*, v. 4, pp. 3639–3644, Oct 2003. doi: 10.1109/ICSMC.2003.1244454.

[19] ARIMOTO, S. "Intelligent Control of Multi-fingered Hands", *Ann. Rev. Contr.*, v. 28, n. 1, pp. 75–85, 2004. ISSN: 1367-5788.

[20] ZHAO, Y., CHEAH, C. C. "Neural Network Control of Multifingered Robot Hands Using Visual Feedback", *IEEE Trans. Neur. Net.,*, v. 20, n. 5, pp. 758–767, May 2009. doi: 10.1109/TNN.2008.2012127.

[21] CHAUMETTE, F., HUTCHINSON, S. "Visual Servoing and Visual Tracking". In: Siciliano, B., Khatib, O. (Eds.), *Springer Handbook of Robotics*, Springer Berlin Heidelberg, pp. 563–583, 2008. ISBN: 978-3-540-23957-4.

[22] SHIRAI, Y., INOUE, H. "Guiding a Robot by Visual Feedback in Assembling Tasks", *Pattern Recognition*, v. 5, n. 2, pp. 99–108, 1973. ISSN: 0031-3203.

[23] SEITZ, M. "Towards Autonomous Robotic Servicing: Using an Integrated Hand-arm-eye System for Manipulating Unknown Objects", *Robot. and Auton. Syst.*, v. 26, n. 1, pp. 23–42, 1999. ISSN: 0921-8890.

[24] LIPPIELLO, V., RUGGIERO, F., SICILIANO, B., et al. "Visual Grasp Planning for Unknown Objects Using a Multi-Fingered Robotic Hand", *IEEE/ASME Trans. on Mech.*, v. 18, n. 3, pp. 1050–â€"1059, 2013.

[25] PRATS, M., MARTINET, P., DEL POBIL, A., et al. "Robotic execution of everyday tasks by means of external vision/force control", *Intelligent Service Robotics*, v. 1, n. 3, pp. 253–266, 2008. ISSN: 1861-2776.

[26] HONDA, K., HASEGAWA, T., KIRIKI, T., et al. "Real-time Pose Estimation of an Object Manipulated by Multi-fingered Hand using 3D Stereo Vision and

Tactile Sensing". In: *Proc. of the IEEE/RSJ Int. Conf. on Intell. Robot. and Syst.*, v. 3, pp. 1814–1819, Oct 1998.

[27] NAMIKI, A., NAKABO, Y., ISHII, I., et al. "High Speed Grasping using Visual and Force Feedback". In: *Proc. of the IEEE Int. Conf. on Rob. and Aut.*, v. 4, pp. 3195–3200, 1999.

[28] YOKOKOHJI, Y., SAKAMOTO, M., YOSHIKAWA, T. "Vision-aided Object Manipulation by a Multifingered Hand with Soft Fingertips". In: *Proc. of the IEEE Int. Conf. on Rob. and Aut.*, v. 4, pp. 3201–3208, 1999.

[29] KAWAMURA, A., TAHARA, K., KURAZUME, R., et al. "Robust Visual Servoing for Object Manipulation Against Temporary Loss of Sensory Information Using a Multi-Fingered Hand-Arm", *Jour. of Robot. and Mech.*, v. 25, n. 1, pp. 125–135, 2013.

[30] JARA, C., POMARES, J., CANDELAS, F., et al. "Optimal Control for Robot-hand Manipulation of an Object using Dynamic Visual Servoing". In: *Proc. of the IEEE/RSJ Int. Conf. on Intell. Robot. and Syst.*, pp. 89–94, Sep 2014.

[31] WEN, J.-Y., WILFINGER, L. "Kinematic Manipulability of General Constrained Rigid Multibody Systems", *IEEE Trans. Rob. Aut.*, v. 15, n. 3, pp. 558–567, Jun 1999. ISSN: 1042-296X. doi: 10.1109/70.768187.

[32] CACCAVALE, F., UCHIYAMA, M. "Cooperative Manipulators". In: Siciliano, B., Khatib, O. (Eds.), *Springer Handbook of Robotics*, Springer Verlag, pp. 701–718, 2008. ISBN: 978-3-540-23957-4. doi: 10.1007/978-3-540-30301-5_30.

[33] MURRAY, R. M., SASTRY, S. S., ZEXIANG, L. *A Mathematical Introduction to Robotic Manipulation.* CRC Press, Inc., 1994.

[34] ZACHI, A., HSU, L., LIZARRALDE, F. "Performing Stable 2D Adaptive Visual Positioning/Tracking Control without Explicit Depth Measurement". In: *Proc. of the IEEE Int. Conf. on Rob. and Aut.*, v. 3, pp. 2297–2302, Apr 2004.

[35] ROBOTIQ. *Specification sheet for the 3-Finger Adaptive Robot Gripper*, Aug 2013. http://robotiq.com/products/industrial-robot-hand/.

[36] M.R.ANDERSEN, T.JENSEN, P.LISOUSKI, et al. *Kinect Depth Sensor Evaluation for Computer Vision Applications.* Relatório técnico, February 2012. Disponível em: <http://eng.au.dk/fileadmin/DJF/ENG/PDF-filer/Tekniske_rapporter/Technical_Report_ECE-TR-6-samlet.pdf>.

[37] REIS, M. F., LEITE, A. C., LIZARRALDE, F., et al. "Kinematic modeling and control design of a multifingered robot hand". In: *Industrial Electronics (ISIE), 2015 IEEE 24th International Symposium on*, pp. 638–643, June 2015. doi: 10.1109/ISIE.2015.7281543.

[38] REIS, M. F., LEITE, A. C., FROM, P. J., et al. "Visual Servoing for Object Manipulation with a Multifingered Robot Hand*", *IFAC-PapersOnLine*, v. 48, n. 19, pp. 1 – 6, 2015. ISSN: 2405-8963. doi: http://dx.doi.org/10.1016/j.ifacol.2015.12.001. Disponível em: <http://www.sciencedirect.com/science/article/pii/S2405896315026257>. 11th {IFAC} Symposium on Robot Control {SYROCO}, Salvador, Brazil, 26-28 August 2015.

[39] REIS, M. F., LEITE, A. C., LIZARRALDE, F. "Modeling and control of a multifingered robot hand for object grasping and manipulation tasks". In: *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pp. 159–164, Dec 2015. doi: 10.1109/CDC.2015.7402102.

[40] COTTLE, R. W. "Manifestations of the Schur complement", *Linear Algebra and its Applications*, v. 8, n. 3, pp. 189 – 211, 1974. ISSN: 0024-3795. doi: http://dx.doi.org/10.1016/0024-3795(74)90066-4. Disponível em: <http://www.sciencedirect.com/science/article/pii/0024379574900664>.

Appendix A

Proofs of Theorems

Here we present the proofs of the theorems presented in chapters 2 and 3.

A.1 Finger kinematic constraints

If the fingertips pose does not depend directly on θ_{ka}, then J^k_{fa} is null. Thus, equation 2.13 reduces to

$$v_k = J^k_{fp}\, J^k_c\, \dot{\theta}_{ka}\,,$$

which is just equation 2.15 with $J^k_f = J^k_{fp}$ and $\tilde{J}^k_c = J^k_c$.

A.2 Finger control

Substituting the control law u into the error state dynamics (3.32), we obtain the following linear error equation:

$$\dot{\tilde{\xi}} + \Lambda\, \tilde{\xi} = 0\,,$$

Choosing Λ as a positive-definite matrix, the error system is asymptotically stable and error state vector $\tilde{\xi}$ tends asymptotically to zero. The convergence to zero of the error state vector $\tilde{\xi}$ ensures the asymptotic convergence of the selected directions of h_1, h_2, h_3 to its desired values h_{1d}, h_{2d}, h_{3d}.

A.3 Prehension control

Substituting the control law u into the error state dynamics (3.27) and applying the conditions of the theorem, we obtain the linear error equation:

$$\dot{\tilde{\xi}}_r + \Lambda_r \tilde{\xi}_r = 0\,,$$

Choosing Λ_r as a positive-definite matrix, the error system is asymptotically stable and the error state vector $\tilde{\xi}_r$ tends asymptotically to zero, ensuring the asymptotic convergence of the selected directions of the relative positions p_r.

A.4 Object manipulation control

Substituting the control law u into the error state dynamics (3.32) and applying the conditions of the theorem, we obtain the following linear error equation:

$$\begin{bmatrix} \dot{\tilde{\xi}}_o \\ \dot{\tilde{\xi}}_r \end{bmatrix} + \begin{bmatrix} \Lambda_o & 0 \\ 0 & \Lambda_r \end{bmatrix} \begin{bmatrix} \tilde{\xi}_o \\ \tilde{\xi}_r \end{bmatrix} = 0\,.$$

Choosing $\Lambda_o \in \mathbb{R}^{m_o \times m_o}$ and $\Lambda_r \in \mathbb{R}^{m_r \times m_r}$ as positive-definite matrices, the error system is asymptotically stable, and the error state vectors $\tilde{\xi}_o$ and $\tilde{\xi}_r$ tend asymptotically to zero, ensuring both the convergence of the state position of the object to its desired values and the convergence of the selected directions of the relative positions p_r to the initial relative state ξ_{r0}.

A.5 HVS control

Substituting the control law 3.41 into the error state dynamics (3.40), we obtain the following linear error equation:

$$\begin{bmatrix} \dot{\tilde{\xi}}_z \\ \dot{\tilde{\xi}}_r \end{bmatrix} + \begin{bmatrix} \Lambda_z & 0 \\ 0 & \Lambda_r \end{bmatrix} \begin{bmatrix} \tilde{\xi}_z \\ \tilde{\xi}_r \end{bmatrix} = 0\,.$$

which is asymptotically stable under the conditions of the theorem and thus, the error state vector $\tilde{\xi}_z$ tends asymptotically to zero. This ensures the asymptotic convergence of the controlled generalized coordinates of $\tilde{p}_{v_{\mathcal{H}}}$ and the controlled relative coordinates of ξ_r to its desired values.

A.6 IBVS control

Substituting the control law 3.49 into the error state dynamics (3.48), we obtain the following linear error equation:

$$\begin{bmatrix} \dot{\tilde{\xi}}_v \\ \dot{\tilde{\xi}}_r \end{bmatrix} + \begin{bmatrix} \Lambda_v & 0 \\ 0 & \Lambda_r \end{bmatrix} \begin{bmatrix} \tilde{\xi}_v \\ \tilde{\xi}_r \end{bmatrix} = 0\,.$$

which is asymptotically stable under the conditions of the theorem and thus, the error state vector $\tilde{\xi}_v$ tends asymptotically to zero. This ensures the asymptotic convergence of the controlled image-space coordinates of ξ_v and the controlled relative coordinates of ξ_r to its desired values.

A.7 Stacked Jacobian singularities

The Moore-Penrose pseudo-inverse matrix of J in terms of J_1 and J_2 is

$$\begin{aligned} J^\dagger = \begin{bmatrix} J_1 \\ J_2 \end{bmatrix}^\dagger &= \begin{bmatrix} J_1^\mathsf{T} & J_2^\mathsf{T} \end{bmatrix} \left(\begin{bmatrix} J_1 \\ J_2 \end{bmatrix} \begin{bmatrix} J_1^\mathsf{T} & J_2^\mathsf{T} \end{bmatrix} \right)^{-1} \\ &= \begin{bmatrix} J_1^\mathsf{T} & J_2^\mathsf{T} \end{bmatrix} \begin{bmatrix} J_1\,J_1^\mathsf{T} & J_1\,J_2^\mathsf{T} \\ J_2\,J_1^\mathsf{T} & J_2\,J_2^\mathsf{T} \end{bmatrix}^{-1} \end{aligned} \tag{A.1}$$

If the last matrix in the right hand side of (A.1) is full rank, than its inverse exists and $J^\dagger$ is uniquely defined. To check out the rank of this matrix, we make use of an important result from the theory of block matrices.

If M is a square block matrix defined in terms of sub-matrices A, B, C and D as:

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix},$$

where the dimensions of these sub-matrices are consistent, than the inverse of M is given by:

$$M^{-1} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} (A - B\,D^{-1}\,C)^{-1} & -A^{-1}\,B\,(D - C\,A^{-1}\,B)^{-1} \\ -C^{-1}\,C\,(A - B\,D^{-1}\,C)^{-1} & (D - C\,A^{-1}\,B)^{-1} \end{bmatrix}$$

where the matrix expressions $(A - B\,D^{-1}\,C)$ and $(D - C\,A^{-1}\,B)$ are known as the Schur complements of M (COTTLE, 1974).

Back to (A.1), if $A = J_1\, J_1^{\mathsf{T}}$, $B = J_1\, J_2^{\mathsf{T}}$, $C = J_2\, J_1^{\mathsf{T}}$ and $D = J_2\, J_2^{\mathsf{T}}$, after some algebra, we get:

$$\begin{bmatrix} J_1\, J_1^{\mathsf{T}} & J_1\, J_2^{\mathsf{T}} \\ J_2\, J_1^{\mathsf{T}} & J_2\, J_2^{\mathsf{T}} \end{bmatrix}^{-1} = \begin{bmatrix} (J_1\, P_{J_2}^{\perp}\, J_1^{\mathsf{T}})^{-1} & -(J_1^{\mathsf{T}})^{+}\, J_2^{\mathsf{T}}\, (J_2\, P_{J_1}^{\perp}\, J_2^{\mathsf{T}})^{-1} \\ -(J_2^{\mathsf{T}})^{+}\, J_1^{\mathsf{T}}\, (J_1\, P_{J_2}^{\perp}\, J_1^{\mathsf{T}})^{-1} & (J_2\, P_{J_1}^{\perp}\, J_2^{\mathsf{T}})^{-1} \end{bmatrix} \tag{A.2}$$

where $P_{J_1}^{\perp} = (I - J_1^{\dagger}\, J_1)$ and $P_{J_2}^{\perp} = (I - J_2^{\dagger}\, J_2)$ are the projection matrices defined previously and

$$\begin{aligned} (J_1^{\mathsf{T}})^{+} &= (J_1\, J_1^{\mathsf{T}})^{-1}\, J_1\,, & J_1^{+}\, J_1 = I \\ (J_2^{\mathsf{T}})^{+} &= (J_2\, J_2^{\mathsf{T}})^{-1}\, J_2\,, & J_2^{+}\, J_2 = I \end{aligned}$$

are known as the Moore-Penrose *left* pseudo-inverse matrix of J_1^{T} and J_2^{T}, in contrast with the *right* Moore-Penrose pseudo-inverses

$$\begin{aligned} J_1^{\dagger} &= J_1^{\mathsf{T}}\, (J_1\, J_1^{\mathsf{T}})^{-1}\,, & J_1\, J_1^{\dagger} = I \\ J_2^{\dagger} &= J_2^{\mathsf{T}}\, (J_2\, J_2^{\mathsf{T}})^{-1}\,, & J_2\, J_2^{\dagger} = I\,. \end{aligned}$$

Note that, from (A.2), the inverse of $J\, J^{\mathsf{T}}$ is only defined if J_1, J_2, $P_{J_1}^{\perp}$ and $P_{J_2}^{\perp}$ are all full rank, and thus, the right pseudo-inverse of J is defined by (A.1). This fact means that J must be full rank as well, what concludes the proof.

Appendix B

Derivation of the active-to-passive joint map in the case of non-integrable kinematic constraints

If it is not possible to obtain 2.10 from 2.9 or if $c_k(.)$ is difficult to obtain by empirical means, equation 2.9 can be differentiated with respect to time, yielding:

$$J_g^k(\theta_k)\,\dot{\theta}_k = 0\,, \qquad J_g^k = \frac{\partial g_k(\theta_k)}{\partial \theta_k}\,, \tag{B.1}$$

where J_g^k is the general constraint Jacobian matrix of each finger. Now, if we use the definition of the vector of joint variables θ_k in terms of the active and passive joints, we can also rewrite B.1 partitioning the general constraint Jacobian matrix J_g^k into its active and passive parts:

$$\left[\begin{array}{cc} J_{ga}^k & J_{gp}^k \end{array}\right] \left[\begin{array}{c} \dot{\theta}_{ka} \\ \dot{\theta}_{kp} \end{array}\right] = J_{ga}^k\,\dot{\theta}_{ka} + J_{gp}^k\,\dot{\theta}_{kp} = 0 \tag{B.2}$$

Then, solving for $\dot{\rho}_k$, we obtain:

$$\dot{\theta}_{kp} = -{J_{gp}^k}^{-1}\,J_{ga}^k\,\dot{\theta}_{ka}\,. \tag{B.3}$$

Now, we focus on the consequences of the following hypothesis about the differential kinematics equations of a general robot finger:

H3 *Both J^k_{ga} and J^k_{gp} matrices are functions of θ_{ka} (the active joints) only.*

Under assumption (H3), we can integrate B.3 directly, obtaining:

$$\theta_{kp} = \theta_{kp_0} + \int_{\theta_{ka_0}}^{\theta_{ka}} -{J^k_{gp}}^{-1} J^k_{ga} \, \mathrm{d}\theta_{ka} \tag{B.4}$$

Note that, if the integral in B.4 can be solved analytically and the initial passive joint angles ρ_{k0} are known, these equations are exactly the constraint equations in 2.10.

Comparing 2.11 and B.3, we note that, in the case of assumption (H3):

$$J^k_c(\alpha_k) = -{J^k_{gp}}^{-1} J^k_{ga} \, . \tag{B.5}$$

This methodology shows that there are two different ways of calculating the differential constraints for the fingers. The first way is to have the set of m_k general constraint equations in 2.9, and then differentiate then with respect to time, obtaining B.2. The other way is to obtain the constraint equations empirically, directly in the form of 2.10 and then perform the time differentiation, directly obtaining 2.11.

In particular, if assumption (H3) is satisfied and if the matrix $-{J^k_{gp}}^{-1} J^k_{ga}$ is integrable, the equivalence between the two methods is easily seen, because one can obtain the constraint relation between the passive and the active joints by direct integration, in B.4. The closed-form solution for the integral will be the desired $c_k(.)$ functions, which must be equivalent to the empirical relations found.